OmegaT 2.6 - User's Guide

Vito Smolej

OmegaT 2.6 - User's Guide

Vito Smolej

Publication date

Abstract

This document is the official user's guide to OmegaT, the free Computer Aided Translation tool. It also contains installation instructions.

Table of Contents

1. About OmegaT - introduction 1
    1. OmegaT highlights 1
    2. Summary of chapters 1
2. Learn to use OmegaT in 5 minutes! 3
    1. Set up a new project 3
    2. Translate the file 3
    3. Validate your tags 3
    4. Generate the translated file 4
    5. Few more things to remember 4
3. Installing and running OmegaT 5
    1. Windows Users 5
    2. Linux (Intel) Users 6
    3. Mac OS X Users 7
    4. Other Systems 8
    5. Using Java Web Start 9
    6. Starting OmegaT from the command line 9
    7. Building OmegaT From Source 14
4. The user interface 15
    1. Main OmegaT window, other windows and dialogs 15
    2. OmegaT main window 16
    3. Other windows 21
5. Menu and Keyboard shortcuts 25
    1. Main Menu 25
    2. Keyboard shortcuts 31
6. Project properties 34
    1. Properties dialog 34
7. File Filters 38
    1. File filters dialog 38
    2. Filter options 38
    3. Edit filter dialog 40
8. OmegaT Files and Folders 43
    1. Translation project files 43
    2. User settings files 45
    3. Application files 46
9. Files to translate 48
    1. File formats 48
    2. Other file formats 49
    3. Right to left languages 50
10. Editing behavior 52
11. Working with plain text 54
    1. Default encoding 54
    2. The OmegaT solution 54
12. Working with formatted text 55
    1. Formatting tags 55
    2. Tag operations 55
    3. Tag group nesting 56
    4. Tag group overlapping 56
    5. Tag validation options 56
    6. Tag group validation 57
    7. Hints for tags management 58
13. Translation memories 59
    1. Translation memories in OmegaT 59
    2. Reusing translation memories 61
    3. Sources with existing translations 63
    4. Pseudo-translated memory 64
    5. Upgrading translation memories 64
14. Source segmentation 65

1. Segmentation rules 65
2. Rule priority 66
3. Creating a new rule 66
4. A few simple examples 66
15. Searches 67
1. Search window 67
2. Using wild cards 68
3. Search methods and options 68
4. Search results display 69
5. Filter entries in editor according to search 69
16. Regular expressions 70
1. Regex tools and examples of use 72
17. Dictionaries 74
1. How to download and install dictionaries 74
2. Problems with dictionaries 75
18. Glossaries 76
1. Usage 76
2. File format 77
3. How to create glossaries 77
4. Using Trados MultiTerm 78
5. Common glossary problems 78
19. Machine Translation 80
1. Introduction 80
2. Google Translate 80
3. OmegaT users and Google Translate 80
4. Belazar 81
5. Apertium 81
6. Machine translation - trouble shooting 82
20. Spell checker 83
1. Installing spelling dictionaries 83
2. Using spelling dictionaries 84
3. Hints 84
21. Miscellaneous subjects 86
1. OmegaT Console Mode 86
2. Automatic Java Properties Aligner 87
3. Font settings 87
4. Preventing data loss 88
A. Languages - ISO 639 code list 89
B. Keyboard shortcuts in the editor 94
C. OmegaT Team Projects 96
1. Version control - introduction 96
2. Sharing a project using SVN 96
3. Using the team project in OmegaT 100
D. Tokenizer plugin 102
1. Introduction 102
2. Installation and command line use 102
3. Mac OS X specifics 102
4. Troubleshooting 103
E. LanguageTool plugin 104
1. Introduction 104
2. Installation and Use 104
F. Scripting plugin 105
1. Introduction 105
2. Installation and Use 105
G. OmegaT on the web 107
1. OmegaT sites and OmegaT SourceForge project 107
2. Bug reports 107
3. Contributing to OmegaT project 107
H. Shortcuts customization 108
1. Shortcuts customization 108

2. Project Menu 109
3. Edit Menu 109
4. GoTo Menu 110
5. View Menu 110
6. Tools Menu 110
7. Options Menu 111
8. Help Menu 111
I. Legal notices 112
1. For the documentation 112
2. For the application 112
J. Acknowledgements 113
1. Thank you all! 113
Index 114

List of Figures

4.1. OmegaT main window 16
4.2. Matches pane 18
4.3. Matches pane setup 19
4.4. multi-word entry in the glossary 20
4.5. Comments pane 21
4.6. Tag validation window 22
4.7. project statistics 23
4.8. Match statistics 24
6.1. Project properties window 35
8.1. OmegaT project 43
8.2. OmegaT projects and subfolders 44
10.1. Editing behavior options 52
12.1. Tag validation entry 58
15.1. Text search Window 67
16.1. Regex Tester 72
17.1. Merriam Webster dictionary - use 74
18.1. Glossary pane 76
18.2. multiple words entries in glossaries - example 77
19.1. Google Translate - example 81
20.1. Spellchecker setup 83
20.2. Using spellchecker 84
E.1. The LanguageTool in OmegaT 104

List of Tables

4.1. Main OmegaT window 15
4.2. Other windows 15
4.3. Settings dialogs 15
4.4. Pane widgets 16
4.5. Main Window - counters 17
4.6. Match pane setup 19
5.1. Main Menu 25
5.2. Project menu 25
5.3. Copy/cut/paste shortcuts 26
5.4. Edit menu 26
5.5. Go To menu 28
5.6. View menu 29
5.7. Tools menu 29
5.8. Options menu 30
5.9. Help menu 31
5.10. Project management shortcuts 31
5.11. Editing shortcuts 32
5.12. Moving around shortcuts 33
5.13. Various shortcuts 33
16.1. Regex - Flags 70
16.2. Regex - Character 70
16.3. Regex - Quotation 70
16.4. Regex - Classes for Unicode blocks and categories 71
16.5. Regex - Character classes 71
16.6. Regex - Predefined character classes 71
16.7. Regex - Boundary matchers 71
16.8. Regex - Greedy quantifiers 71
16.9. Regex - Reluctant (non-greedy) quantifiers 72
16.10. Regex - Logical operators 72
16.11. Regex - Examples of regular expressions in translations 73
A.1. ISO 639-1/639-2 Language code list 89
B.1. Key behavior in the editor 94
H.1. Project Menu 109
H.2. Edit Menu 109
H.3. GoTo Menu 110
H.4. View Menu 110
H.5. Tools Menu 110
H.6. Options Menu 111
H.7. Help Menu 111

Chapter 1. About OmegaT - introduction

1. OmegaT highlights

OmegaT is a free multi platform Computer Aided Translation tool, with the following highlights:

- **Translation memory:** OmegaT stores your translations in a translation memory file. At the same time, it can use memories files from previous translations for reference. Translation memories can be very useful in a translation where there are numerous repetitions or reasonably similar segments of text. OmegaT uses translation memories to store your previous translations and then suggest likely translations for the text you are currently working on.

  These translation memories can be very useful when a document that has already been translated needs to be updated. Unchanged sentences are automatically translated, while updated sentences are shown with the translation of the most similar, older sentence. Modifications to the original document are thus handled with greater ease. If you are supplied with previously created translation memories , for example by your translation agency or your client, OmegaT is able to use these as reference memories.

  OmegaT uses the standard tmx file format to store and access translation memories, which guarantees that you can exchange your translation material with other CAT applications supporting this file format.

- **Terminology management**: Terminology management is important for translation consistency. OmegaT uses glossaries containing translations of single words or small phrases: a simplified bilingual dictionary for a specific domain. For your reference, OmegaT displays the translation of any word that is both present in the segment and registered in the glossary.

- **Translation process:** Imagine having to translate something; from a single file to a folder containing subfolders each with a number of files in a variety of formats. When you let OmegaT know the files that you need to translate, it looks for the files it understands based on file filtering rules, recognizes the textual parts within them, splits up the text groups according to the segmentation rules, and displays the segments one by one so that you can proceed with the translation. OmegaT stores your translations and proposes possible translations from similar segments in registered in the translation memory files. When you are ready to view the final product, you can export the translated files, open them in the appropriate application and view the translation in the final format...

2. Summary of chapters

This documentation is intended both as a tutorial and as a reference guide. Here is a short summary of the chapters and their contents.

- **Learn to use OmegaT in 5 minutes!:** this chapter is intended as a quick tutorial for beginners as well as people who already know CAT tools, showing the complete procedure from opening a new translation project through to completing the translation.

- **Installing and running OmegaT:** this chapter is useful when you first begin using OmegaT. It contains the specific instructions on how to install OmegaT and run it on Windows, Mac OS X and Linux. For advanced users, the chapter describes the command line mode and its possibilities.

- **The user interface**,**Main Menu and Keyboard Shortcuts**: these two chapters are likely to be heavily consulted, since they explain the user interface of OmegaT and the functions available via the main menu and the keyboard shortcuts.
- **Project properties**, **OmegaT Files and Folders:** a project in the context of OmegaT is the piece of work that OmegaT as a CAT tool is able to handle. This chapter describes the project properties, such as the source and target languages. The second of these chapters describes the various subfolders and files in a translation project and their role as well as other user and application files associated with OmegaT.
- **Editing field behavior:** a short chapter describing how to set up the editing behavior of the segment being translated.
- **Working with plain text** and **Working with formatted text**: these two chapters explain certain important points concerning texts to be translated, such as the encoding set (in the case of plain text files) and tag handling (in the case of formatted text).
- **Translation memories:** explains the role of the various subfolders containing translation memories, and provides information on other important aspects relating to translation memories.
- **Segmentation**:translation memory tools work with textual units called segments. In OmegaT, segments can be based on paragraphs or on segmentation rules. Paragraph segmentation is less frequently used, but can be useful in cases of so-called "creative" texts. The rule-based segmentation is usually synonymous with sentence-based segmentation. A number of rule sets are provided while additional rules can be defined by the user, as described in this chapter.
- **Searches** and **Regular expressions**: Searches in OmegaT can be as simple as "*list segments with the word* '*kangaroo*' ". They can also be complex, allowing for instance to search for segments with two or more consecutive spaces. In this case the regular expression ^s^s+ would be used to find and list the offending segments. Regular expressions are also used extensively in the segmentation rules.
- **Dictionaries, Glossaries**, **Machine translation** , **Spellchecker, LanguageTool:** OmegaT supports an extensive use of dictionaries and glossaries. If an Internet connection is available, various MT services such as Google Translate and Microsoft Translator can be used from within OmegaT. If spell checking is activated, spelling mistakes and typos are recognized and can be corrected during translation. The open source LanguageTool can be used to correct common grammatical and stylistic mistakes.
- **Miscellanea**: deals with other issues of interest, such as how to avoid losing data.
- **Appendices** contain the following information
  - **OmegaT on the web:** information regarding on-line OmegaT resources
  - **Languages:** the ISO list of languages and language codes is provided
  - **Keyboard shortcuts in the editor**: the list of shortcuts used in the editor
  - **Shortcuts customization:** shortcuts can be customized to your personal preferences
  - **Introduction into Tokenizer and Scripting plugins**
  - **Team Projects**
  - **Legal notices and Acknowledgements**
- **Keyword index:** an extensive keyword index is provided to help the reader find the relevant information.

Chapter 2. Learn to use OmegaT in 5 minutes!

1. Set up a new project

*Note:*On an Apple Mac, use the **Cmd** or **Apple** key instead of the **Ctrl** key

To start using OmegaT, first create a project that will hold all your files, such as your source file, translation memories, glossaries, and eventually your translated file. In the Project menu, select New... and type a name for your project. Remember where you are creating the project, because you will need to return to it later.

After you give your project a name, the Create New Project dialog will open. At the top of that dialog, select your source file's language and the language that your translated file will be, and click OK to continue.

If you are interested in other settings of this dialog, you can return to it any time by pressing **Ctrl+E.**

Next, the Project Files dialog opens. Click the Import source files... to select your source file. OmegaT will then copy the selected file to the /source/subfolder of your newly created project. After the source file has loaded in the Editor pane, you can close the Project files dialog.

2. Translate the file

OmegaT will present one segment at a time for you to translate. After you have translated each segment, press **Ctrl+U** to move to the next untranslated segment. Whenever you want to see what your translation will look like in its final format, press **Ctrl+D** to generate the translated document, which will be created in the /target/ subfolder of your project folder. During translation, use the Edit and Go To menus to perform various useful functions.

3. Validate your tags

If your source file is a formatted file, e.g. Microsoft Word, LibreOffice Writer or HTML, OmegaT will convert the formatting into tags that surround the text that you translate. Often documents will also have tags that have nothing to do with formatting, but which are also important in the source file (and in the translated file). A source sentence might look like

OmegaT is an *easy to use* program
for **eager** translators.

OmegaT, however, will present this sentence in the following fashion:

OmegaT is an <t0/>easy to use<t1/> program
for <t2/>eager<t3/> translators.

The tags in OmegaT are greyed, so they are easy to recognise. They are not protected, so that you can delete them, enter them by hand or move them around in the target sentence. However, if you made mistakes when you typed the formatting tags, your translated file might fail to open. Therefore, press **Ctrl+T** before you generate your translated file, to validate that your tags are correct.

4. Generate the translated file

Once you have made certain that there are no tag errors in your translation, press **Ctrl+D** to generate the target file, which will be created in the /target/subfolder of your project folder.

5. Few more things to remember

- If a file does not loads into the Editor pane, then it could be that your source file is in a format that doesn't work in OmegaT. See Chapter 7: File Filters in the User Manual for a list of file formats that OmegaT can handle.
- You can create a new project for each new job, and you can add several source files to a project at a time.
- To remind yourself of the project's initial settings, open the Project Settings dialog by pressing **Ctrl+E**. To see a list of files in the project, open the Project Files dialog by pressing **Ctrl+L**.
- At the end of your translation, OmegaT exports three translation memories called "level1", "level2" and "omegat" to your project folder. The "level1" and "level2" memories can be shared with users of other translation programs. The memory named "omegat" can be used in OmegaT itself, in future projects that you create. If you place translation memory files in the /tm/ subfolder of your project folder, OmegaT will automatically search them for similar segments, called "fuzzy matches".
- You can create a glossary by pressing Ctrl+Shift+G, or copy existing glossaries to the/ glossary/subfolder of your project folder, and OmegaT will automatically look up words in them.
- It is often useful to search for words and phrases in the source text and in your translation, so press **Ctrl+F** for the "Text Search" dialog at any time.
- For a more comprehensive introduction see OmegaT for beginners [http://www.omegat.org/en/tutorial/OmegaT%20for%20Beginners.pdf] on the OmegaT web site. If you need assistance with any aspect of OmegaT, feel free to join the OmegaT users group. [http://tech.groups.yahoo.com/group/OmegaT/]

Chapter 3. Installing and running OmegaT

1. Windows Users

1.1. Downloading the package

Do you have a Java implementation compatible with Oracle's Java 1.5 JRE?

- **Yes:** download *OmegaT_2.n.n_Windows_without_JRE.exe*.
- **No / I don't know:** download *OmegaT_2.n.n_Windows.exe.*

  This package is bundled with Oracle's Java Runtime Environment. This JRE will not interfere with other Java implementations that may already be installed on your system.

1.2. Installing OmegaT

To install OmegaT, double-click on the program you have downloaded.

At the beginning of the installation you can select the language to be used during the installation. In the following window you can indicate that the language selected is to be used in OmegaT. If you check the corresponding checkbox, the OmegaT.l4J.ini file is modified to use the language selected (see next section for details). Later, after you have accepted the license agreement, the setup program asks you whether you wish to create a folder in the *start* menu, and whether you wish to create a shortcut on the desktop and in the quick launch bar - you can create these shortcuts later by dragging *OmegaT.exe* to the desktop or to the start menu to link it from there. The last frame offers you to have a look at the readme and changes files for the version you have installed.

1.3. Running OmegaT

Once OmegaT is installed, you can click on *OmegaT.jar* to launch it directly or you can launch it directly from the command line.

The simplest way to launch OmegaT, however, is to execute the *OmegaT.exe* program. The options for the program start-up in this case will be read from the *OmegaT.l4J.ini* file, which resides in the same folder as the exe file and which you can edit to reflect your setup. The following example for the INI file reserves 1GB of memory, requests French as the user language and Canada as the country:

```
# OmegaT.exe runtime configuration
# To use a parameter, remove the '#' before the '-'
# Memory
-Xmx1024M
# Language
-Duser.language=FR
# Country
-Duser.country=CA
```

1.4. Upgrading OmegaT

This information applies only to the "Traditional" Windows versions of OmegaT. It does not apply to the Web Start versions, which are upgraded automatically, nor to cross-platform versions installed on Windows.

If you already have a version of OmegaT installed on your PC and wish to upgrade to a more recent version, you have two options:

- **Install over the existing installation.** To do this, simply select the same installation folder as the existing installation when installing the new version. The "old" version of OmegaT will be overwritten, but any settings from it will be retained. This includes preferences set from within OmegaT, any changes you have made to your OmegaT.l4J.ini file, and also your launch script (.bat file), if you are using one.

With this method, you may also download the "Windows without JRE" version, since the new installation will use your existing JRE.

- **Install to a new folder.** This will enable you to keep both versions side-by-side, which you may wish to do until you feel comfortable with the new version. This method will also use preferences and settings you have made from within OmegaT. In this case, however:
  - If you have made changes to your OmegaT.l4J.ini file and/or are using a .bat file, you must copy these over.
  - If your existing OmegaT installation is a "Windows with JRE" version, the new version must also be a "Windows with JRE" version.

2. Linux (Intel) Users

2.1. Downloading the right package

Do you have a Java implementation compatible with Oracle's Java 1.5 JRE?

- **Yes:** download *OmegaT_2.n.n_Without_JRE.zip*.
- **No / I don't know:** download *OmegaT_2.n.n_Linux.tar.bz2.*

  This package is bundled with Oracle's Java Runtime Environment. This JRE will not interfere with other Java implementations that may already be installed on your system.

2.2. Installing OmegaT

Unpack/untar the downloaded file. This will create an omegat/ folder in the working folder in which you will find all the files needed to run OmegaT. To untar the *.tar.gz* file:

$ tar xf downloaded_file.tar.gz

2.3. Adding OmegaT to your menus (KDE) or panels (Gnome)

2.3.1. KDE 4 Users

You can add OmegaT to your menus as follows:

- Press **Alt+F2** to show KRunner. Type *kmenuedit+enter* to run the command. The KMenuEditor appears. In KMenuEditor select *File -> New Item.*
- Then, after selecting a suitable menu, add a submenu/item with *File - New* Submenu and *File - New Item*. Enter OmegaT as the name of the new item.
- In the "Command" field, use the navigation button to find your OmegaT launch script (the file named OmegaT in the unpacked folder), and select it.
- Click on the icon button (to the right of the Name/Description/Comment fields)

- Other Icons - Browse, and navigate to the /images subfolder in the OmegaT application folder. Select the OmegaT.png icon.
- Finally, save the changes with *File - Save.*

2.3.2. GNOME Users

You can add OmegaT to your menus as follows:

- Right-click on the panel - *Add New Launcher.*
- Enter "OmegaT" in the "Name" field; in the "Command" field, use the navigation button to find your OmegaT launch script (the file named OmegaT in the unpacked folder). Select it and confirm with OK.
- Click on the icon button, then hit *Browse...* and navigate to the /images subfolder in the OmegaT application folder. Select the *OmegaT.png* icon. GNOME may fail to display the icon files in the available formats and initially appear to expect an SVG file, but if the folder is selected, the files should appear and OmegaT.png can be selected.

2.4. Running OmegaT

You can launch OmegaT from the command line with a script that includes start-up options or you can click on *OmegaT.jar* to launch it directly. Methods differ depending on the distribution. Make sure that your *PATH* settings are correct and that *.jar* files are properly associated with a Java launcher. Check "Command line launching" below for more information.

3. Mac OS X Users

3.1. Downloading the package

OmegaT requires Java 1.5 to run. Use "Software Update..." from the Apple menu to update your Java version.

Download *OmegaT_2.n.n_Mac.zip.*

3.2. Installing OmegaT

Double click on *OmegaT_2.n.n_Mac.zip* to unpack it. This creates a folder called *OmegaT* . The folder contains 2 files: index.html and *OmegaT.app.* Copy the folder to a suitable folder (e.g. Applications). Once you have done this, you can delete the *OmegaT_2.n.n_Mac.zip* file, it is no longer needed.

3.3. Adding OmegaT to the Dock

Drag and drop *OmegaT.app* onto the Dock.

3.4. Running OmegaT

Double-click on *OmegaT.app* or click on its location in the Dock.

To access *Info.plist*, right-click on *OmegaT.app* and select "Show Package Contents", then open the file by right-clicking on it and selecting your text editor of choice. You can also "cd" there directly from the command line and open *Info.plist* in a command line editor like emacs or vi.

You can modify the *Properties* as well as the *VMOptions* part.

If you wish to add values to *VMOptions*, put a space between them. For example, -Xmx1024M -Duser.language=ja will launch OmegaT with 1024MB of memory and with the user interface in Japanese.

To launch multiple instances of *OmegaT.app*, double-click the file *JavaApplicationStub* located in *OmegaT.app/ Contents/MacOS/*.

Use *OmegaT.jar* file located in *OmegaT.app/Contents/Resources/Java/* to launch OmegaT from the command line. Check "Command line launching" below for more information.

3.5. Mac OS X goodies

OmegaT.app can be accessed from the Mac OS X Services. You can thus select a word anywhere in OmegaT and use Services to check this word, for instance in Spotlight or in Google. You can also use AppleScript or Automator to create Services or scripts that will automate frequent actions

4. Other Systems

This information applies to systems such as Solaris SPARC/x86/x64, Linux x64/PowerPC, Windows x64

4.1. Downloading the right package

OmegaT is available bundled with a Oracle Java JRE for Linux (Intel x86) and Windows platforms. Users of other platforms (Linux PowerPC, Linux x64, Solaris SPARC/x86/x64, Windows x64 etc) must have a running compatible Java JRE on their system to be able to use OmegaT.

Do you have a Java implementation compatible with Oracle's Java 1.5 JRE?

- **Yes:** download *OmegaT_2.n.n_Windows_without_JRE.*zip. This package can be used on any platform where a Java 1.5 JRE compatible JRE is installed.
- **I don't know:** open a terminal and type "java -version". If a "command not found" or similar message is returned, it is likely that Java is not installed on your system
- **No:** obtain a Java JRE for your system (see below) and download *OmegaT_2.n.n_Without_JRE.zip*.

  Oracle provides JREs for Solaris SPARC/x86 (Java 1.5) and for Linux x64, Solaris x64, Windows x64 (Java 1.5) at http://www.oracle.com/technetwork/java/archive-139210.html

  IBM provides JREs for Linux PowerPC at http://www.ibm.com/developerworks/java/jdk/linux/download.htm [http://www.ibm.com/developerworks/java/jdk/linux/download.html]

  Follow the installation instructions of the package you need.

4.2. Installing OmegaT

To install OmegaT, simply unpack the **OmegaT_2.n.n_Without_JRE.zip** file. This creates an *./OmegaT_2.n.n_Without_JRE/* folder in the working folder with all the files necessary to run OmegaT.

4.3. Installing convenient shortcuts

Follow your system's instructions to install OmegaT shortcuts in convenient places of your choosing.

4.4. Running OmegaT

Once OmegaT is installed, you can launch it directly from the command line, you can create a script that includes launch parameters for the command line or you can click on *OmegaT.jar*

to launch it directly. Methods differ depending on the distribution. Make sure that your *PATH* settings are correct and that *.jar* files are properly associated with a Java launcher. Check "Command line launching" below for more information.

5. Using Java Web Start

Java Web Start technology (part of Java 1.5 and above) can be used to deploy standalone Java software applications with a single click over the network. Java Web Start ensures that the latest version of the application will be deployed, as well as the correct version of the Java Runtime Environment (JRE) used. To start OmegaT for the first time with Java Web Start, load the following URL in your browser:

http://omegat.sourceforge.net/webstart/OmegaT.jnlp

Download the file *OmegaT.jnlp* and then click on it. During the installation, depending on your operating system, you may receive several security warnings. The permissions you give to this version (which may appear as "unrestricted access to the computer") are identical to the permissions you give to the local version, i.e., they allow access to the hard drive of the computer. Subsequent clicks on *OmegaT.jnlp* will check for any upgrades, install them, if there are any, and then start OmegaT. After the initial installation you can, of course, also use *OmegaT.jnlp* also when you are offline.

Privacy: OmegaT Java Web Start does not save any of your information beyond the computer on which you are running it. The application runs on your machine only. Your documents and translation memories remain on your computer, and the OmegaT project will have no access to your work or information.

Note that if you need or wish to use any of the launch command arguments (see above), you must use the normal installation.

6. Starting OmegaT from the command line

Normally, it is not necessary to start OmegaT from the command line. However, the command-line alternative allows the user to control and modify the program's behavior. There are two ways of launching OmegaT using the command line.

6.1. Opening a command line window

A command line window is also referred to as a "terminal window". On Windows it is called an "MS-DOS window" and is available from the Start Menu, inside Programs, through the "MS-DOS" item. The Mac OS X equivalent is the application Terminal located in Applications → Utilities.

To launch OmegaT, you must normally type two commands. The first of these is:

cd {folder}

where *{folder}* is the name of the folder, with complete path, in which your OmegaT program - specifically, the file *OmegaT.jar* - is located. In practice, this command will therefore be something like this:

On Windows

cd C:\Program Files\OmegaT

On Mac OS X

cd <OmegaT.app location>/OmegaT.app/Contents/Resources/Java/

On Linux

cd /usr/local/omegat

This command changes the folder to the folder containing the executable OmegaT file. The second command is the command which actually launches OmegaT. In its most basic form, this command is:

java -jar OmegaT.jar

Pay attention to the capitalization - in OS other than Windows, the program will not start, if you enter *omegat* instead of *OmegaT* !

This method has a particular benefit of being suitable for finding causes of problems: if an error occurs during use of the program, an error message is output in the terminal window which may contain useful information on the cause of the error.

The above method somewhat impractical way of launching a program routinely. For this reason, the two commands described above are contained in a file (a "script", also called a ".bat file" on Windows systems).

When this file is executed, the commands within it are automatically carried out. Consequently, to make changes to the launch command, it is sufficient to modify the file.

6.2. Launch command arguments

The basic command has already been mentioned above. Changes to this command involve the addition of "arguments" to it. Arguments are added after the initial "*java*", and before the "*-jar OmegaT.jar*". Note that in Windows you can change the *OmegaT.l4J.ini* file to reflect your preferences. On Mac OS X, you can change the *Info.plist* located in *OmegaT.app/Contents/* to do the same.

A list of possible arguments is given below. Advanced users can obtain more information on the arguments by typing *man java* in the terminal window.

- **User interface language**

  -Duser.language=XX Normally, i.e. when OmegaT is launched without any arguments, the program first detects the language of the user's operating system. If a user interface in this language is available, OmegaT uses it. So, if the user's operating system is Russian and OmegaT has been localized in Russian, OmegaT is displayed with a Russian user interface, Russian menus, etc. If the language of the user's system is not available, OmegaT defaults to English. This is the standard behavior.

  The "*-Duser.language=XX*" argument causes OmegaT to use the language specified rather than the language of the user's operating system. "XX" in the command stands for the two-digit code of the desired language. To launch OmegaT with a French interface (for example on a Russian operating system), the command would therefore be:

  java -Duser.language=fr -jar OmegaT.jar

- **User country**

  -Duser.country=XX Besides the language, you can also specify the country, for example CN or TW in case of the Chinese language. To display the instant start guide in the desired language, you need to specify both the language and the country. This is necessary even if there's only one combination available, like pt_BR in case of Portuguese / Brazil.

- **Font antialiasing** - (Java 1.5 only)

  -Dswing.aatext=true This argument effects anti-aliasing of the fonts, thereby improving their appearance.

- **Memory assignment**

-XmxZZM This command assigns more memory to OmegaT. By default, 512 MB are assigned, so there is no advantage in assigning less than this figure. "ZZ" stands for the amount of memory assigned, in megabytes. The command to launch OmegaT with assignment of 1024 MB (1 gigabyte) of memory is therefore:

java -Xmx1024M -jar OmegaT.jar

- **Proxy host IP address**

  -Dhttp.proxyHost=nnn.nnn.nnn.nnn The IP address of your proxy server, if your system uses a proxy.

- **Proxy host port number**

  -Dhttp.proxyPort=NNNN The port number your system uses to access the proxy server.

- **Google Translate V2**

  -Dgoogle.api.key=A123456789B123456789C123456789D12345678 If you have signed up for the Google Translate services, enter your private Google API key here. Note that the key is 38 characters long.

- **Microsoft Translator**

  Make sure that you have a free Microsoft account. You'll need this to sign-in to Windows Azure Marketplace [http://datamarket.azure.com/dataset/bing/microsofttranslator#schema] and use the Translator service. Note that up to 2M characters per month are free of charge. The two entries required are available in your account page [https://datamarket.azure.com/account] under Primary account key and Customer-ID:

  -Dmicrosoft.api.client_id=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

  -
  Dmicrosoft.api.client_secret=XXXX9xXxX9xXXxxXXX9xxX99xXXXX9xx9XXxXxXXXXX=

- **Team project option**

  -no-team addresses projects set up for team work. Use it if OmegaT is not to synchronize the project contents.

Arguments can be combined: to launch OmegaT with all the examples described above, the command would be:

java -Dswing.aatext=true -Duser.language=pt -Duser.country=BR -Xmx1024M -DDhttp.proxyHost=192.168.1.1 -Dhttp.proxyport=3128 -jar -OmegaT.jar

6.3. OmegaT in the command line mode

The purpose of the console mode is to use OmegaT as a translation tool in a scripting environment. When started in console mode, no GUI is loaded (so it will work on any console) and the given project is automatically processed as requested.

6.3.1. Prerequisites

To run OmegaT in the command line mode, a valid OmegaT project must be present. The location does not matter, since you have to add it to the command line at the start-up anyway.

If you need altered settings, the configuration files must be available. This can be achieved in two ways:

- Run OmegaT normally (with the GUI) and specify the settings. If you start OmegaT in console mode, it will use the same settings.

- If you can't run OmegaT normally (no graphical environment available): copy the settings files from some other OmegaT installation on another machine to a specific folder. The location does not matter, since you can add it to the command line at startup. The relevant files are filters.conf and segmentation.conf and can be found in the user home folder (e.g. C:\Documents and Settings\%User%\OmegaT under Windows, %user%/.omegat/ under Linux)

6.3.2. Starting in console mode

To start OmegaT in console mode, some extra parameters have to be passed to it on startup. The most important is <project-dir>, and optionally --config-dir=<config-dir>. Example:

java -jar OmegaT.jar /path/to/project \

--config-dir=/path/to/config-files/ \

--mode=console-translate|console-createpseudotranslatetmx|console-align

--source-pattern={regexp}

Note that all parameters start with a double - character.

Explanation:

- <project-dir> tells OmegaT where to find the project to translate. If given, OmegaT starts in console mode and will translate the given project.
- --config-dir=<config-dir> tells OmegaT in which folder the configuration files are stored. If not given, OmegaT reverts to default values (OmegaT folder under user home or, if unavailable, the current working folder). Note double - character
- **--***mode=...***-** OmegaT starts in console mode to perform one of the following services automatically
  - --mode=console-translate

    In this mode, OmegaT will attempt to translate the files in /source/ with the available translation memories. This is useful to run OmegaT on a server with TMX files automatically fed to a project.
  - --mode=console-createpseudotranslatetmx

    In this mode OmegaT will create a TMX for the whole project, based on the source files only. You specify the TMX file to be created with

    --pseudotranslatetmx=allsegments.tmx --pseudotranslatetype=[equal|empty]

    The argument *pseudotranslatetype* specifies, whether the target segments are to be equal to the source, or left empty.
  - --mode=console-align

    In this mode, OmegaT will align the Java properties files found in the /source/ folder of the project to the contents found at the specified location. The resulting TMX is stored in the /omegat/ folder under the name align.tmx.

    Additional parameter is required in this case, specifying the location of the target data:

    --alignDir={location of translated files}

    alignDir must contain a translation in the target language of the project. For instance, if the project is EN->FR, alignDir must contain a bundle ending with _fr. The resulting tmx is stored in the omegat folder under the name align.tmx.
- --source-pattern={regexp}

When mode has been used, this option will specify the files to be processed automatically. If the parameter is not specified, all files will be processed. Here's few typical examples to limit your choice:

- *.*\.html*

  All HTML files will be translated - note that the period in the usual *.html has to be escaped (\.) as specified by the rules for regular expressions

- test\.html

  Only the file test.html at the root of the source folder will be translated. If there are other files named test.html in other folders, they will be ignored.

- dir-10\\test\.html

  Only the file test.html in the folder dir-10 will be processed. Again note that the backslash is escaped as well.

- --output-tag-validation-={regexp}

  When mode has been used, this option will specify the files to be processed automatically. If the parameter is not specified, all files will be processed. Here's few typical examples to limit your choice:

  - *.*\.html*

    All HTML files will be translated - note that the period in the usual *.html has to be escaped (\.) as specified by the rules for regular expressions

  - test\.html

    Only the file test.html at the root of the source folder will be translated. If there are other files named test.html in other folders, they will be ignored.

  - dir-10\\test\.html

    Only the file test.html in the folder dir-10 will be processed. Again note that the backslash is escaped as well.

- --tag-validation=[abort|warn] *outputFileName*

  This option allows the tag validation in a batch mode. If abort is selected, the tag validator will stop on the first invalid segment. If warn is specified, the tag validator will process all segments and write warnings about any segments with invalid tags into the file specified.

6.3.3. Quiet option

An extra command line parameter specific to console mode: --quiet. In the quiet mode, less info is logged to the screen. The messages you would usually find in the status bar are not displayed.

Usage: java -jar OmegaT.jar /path/to/project --mode=console-translate --quiet

6.3.4. Tag validation option

Another extra command line parameter specific to console mode: --tag-validation=[abort|warn]. When this parameter is added, tag validation is done prior to translation/aligning. If the value is abort, then on tag errors the errors are printed and the program stops. If the value is warn then the errors are printed but OmegaT continues.

Usage: java -jar OmegaT.jar /path/to/project --mode=console-translate --tag-validation=abort

7. Building OmegaT From Source

Note that you will need the ant program (*http://ant.apache.org/bindownload.cgi*) to build your own version of OmegaT. Unpack the *OmegaT_2.n.n_Source.zip* file and enter the OmegaT_2.n.n_Source folder or enter the *./omegat/* folder of the SVN checked out code. Please make sure that a build.xml file is present in that folder. Then, on the command line, type:

$ ant jar release

This will create a full distribution of OmegaT in the ./dist/ folder, where you will find all the files necessary to run OmegaT.

Chapter 4. The user interface

1. Main OmegaT window, other windows and dialogs

OmegaT main window contains the main menu, status bar and several panes. Additional windows are available, as well as dialogs, used to change OmegaT project settings. The information below summarizes their use and how they are invoked:

Table 4.1. Main OmegaT window

Editor pane	where you type and edit the translation
Match pane	displays the most similar segments from translation memories
Glossary pane	displays terminology found for items in the current segment
Dictonary pane	displays dictionary hits found for items in the current segment
Machine Translations pane	displays the translation, provided by MT services
Notes pane	notes pertaining to the current segment, e.g. alternative translations or the current key in case of key=value file formats
Comments pane	Comments by the author in PO files, or the name of the attribute being translated (in XHTML)

Table 4.2. Other windows

Tag Validation window	Used to validate tags (open with **Ctrl**+**T**, close with **Esc**)
Help browser	Used to display the user manual (open with **F1**, close with **Esc)**
Statistics window	Used to open the window with the statistics of the project, display it, using Tools → Statistics.
Match statistics window	Used to display the match statistics of the project, select Tools → Match statisticsto open it.

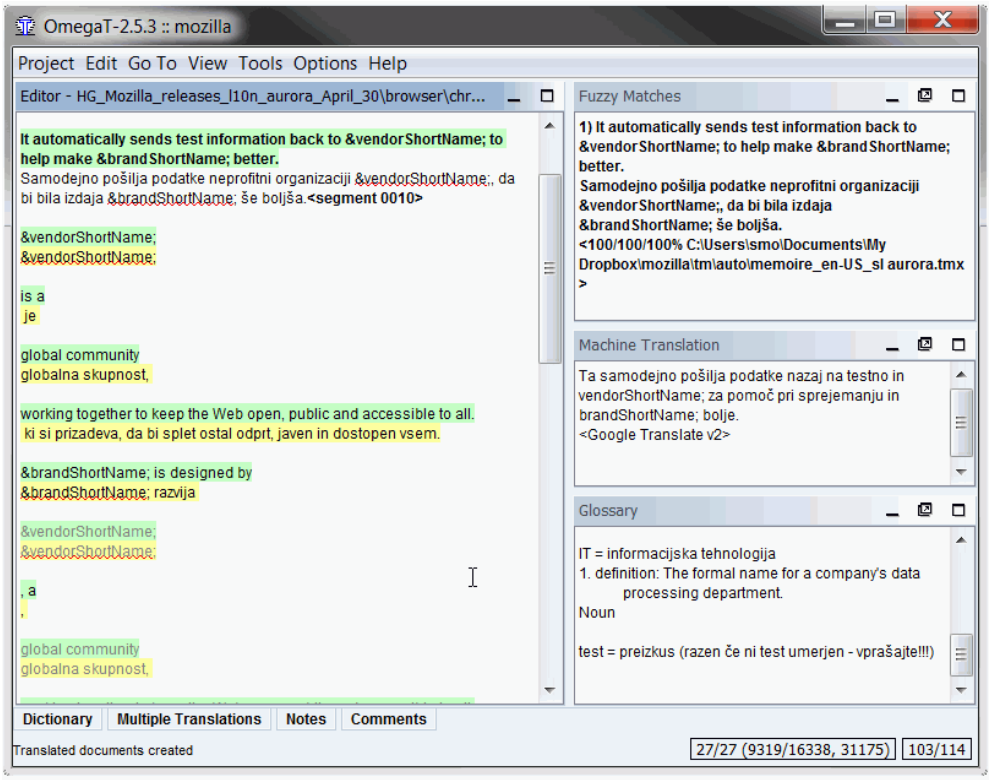
Table 4.3. Settings dialogs

Project properties	Used to modify the project folders and languages (access via Ctrl+E shortcut or Project → properties..., close via **Esc)**
Font	Used to modify the font used by OmegaT to display source, translation, matches and glossary terms, (access via Options → Font..., close via **Esc**)
File filters	Used to adjust the handling of supported file formats (access via Options → File Filters..., close via **Esc**)

Segmentation	Used to change the way your text is segmented into sentences (access via Options → Segmentation, close via **Esc**)
Editing Behavior	Used to change how OmegaT behaves when you iterate between the segments (access via Options → Behavior..., close via **Esc**)

2. OmegaT main window

Figure 4.1. OmegaT main window



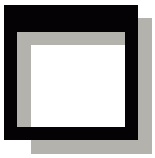
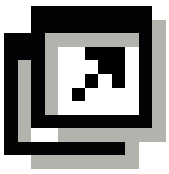
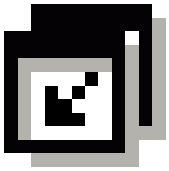
The main window consists of several panes, the main menu and a status bar. You can change the position of any pane or even undock it to a separate window by clicking and dragging the pane by its name. Depending on the pane status, different signs can appear at its top right corner:

Note

If you can not see all the panes (be it opened or minimized), pressing Options > Restore Main Window will restore them to the state, defined in the installation.

Table 4.4. Pane widgets

	minimizes the pane, so that only its name is shown at the bottom of the window

	maximizes the pane
	restores the layout before the maximizing step
	undocks the pane from the main window
	puts the pane back within the main window

You can overlap panes if desired. When this is done the panes display a tab at the top. The separators between the panes can be dragged to resize panes. Should you lose track of your changes to the user interface, you can use Options → Restore the main window any time to return to the original layout.

The counters in the lower right corner keep track of the progress of the translation (numbers in the left hand column refer to the figure above):

Table 4.5. Main Window - counters

27/27	number of segments - translated vs total for the current file
9319/16338	number of unique segments - translated vs total in the project
31175	total number of segments (including repeats) in the project
103/114	number of source and target characters in the current segment

From a practical point of view, the most important pair of numbers is the second pair: it tells, how much you have done so far, in relation to the total or second number. The project in the example is evidently finished, as all the unique segments have been translated.

2.1. Editor pane

This is where you type and edit your translation. The Editor pane displays the text of the partially translated document: the text already translated is displayed in translation while the untranslated text is displayed in the original language. The displayed text is split into segments and you may scroll through the document and double-click on any segment to open and edit it. In the above case, the segments already translated are shown in yellow.

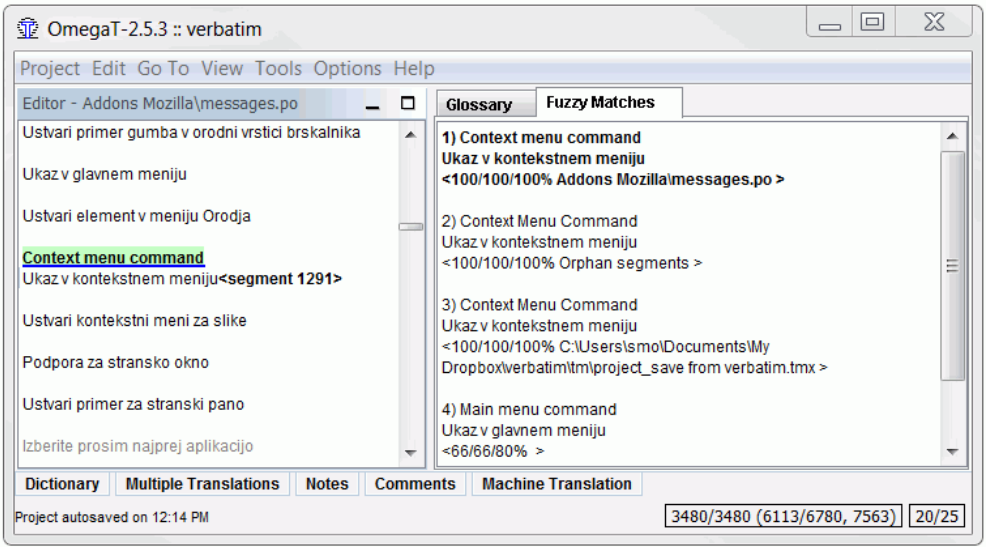
One of the above segments is the current segment. It is the segment that is displayed in two parts. The upper part is in the source language, in bold characters with a green background color, the lower part is the editing field, ended by a marker: the marker is <segment nnnn> where nnnn is a number of the segment in the project. Use the upper part as a reference and replace or modify the contents of the editing field with your translation.

Depending upon the preferred editing behavior, the editing field for the untranslated segment may be empty, contain the source text, or contain the translation of the string most similar to the one to be translated. When you move to another segment, the translation is validated and stored. If you want the translation to be the same as the source, simply make the editing field empty by removing all the text (select all with **Ctrl+A** and delete with **Del**). OmegaT is able to store translations that are identical to the source. This is useful for documents that contain trade marks, names or other proper nouns, or parts in a third language that do not require translation. See *Translation editing* for more details.

If you right click on the Editor pane, a pop-up menu opens, offering **Cut, Copy, Paste** (i.e. same functions as **Ctrl**+**X**, **Ctrl**+**C** and **Ctrl**+**V**) and the **GoTo segment** functions.

2.2. Fuzzy matches pane

Figure 4.2. Matches pane


OmegaT-2.5.3 :: verbatim
Project Edit Go To View Tools Options Help
Editor - Addons Mozilla\messages.po
Ustvari primer gumba v orodni vrstici brskalnika
Ukaz v glavnem meniju
Ustvari element v meniju Orodja
Context menu command
Ukaz v kontekstnem meniju<segment 1291>
Ustvari kontekstni meni za slike
Podpora za stransko okno
Ustvari primer za stranski pano
Izberite prosim najprej aplikacijo
Glossary
Fuzzy Matches
1) Context menu command
Ukaz v kontekstnem meniju
<100/100/100% Addons Mozilla\messages.po >
2) Context Menu Command
Ukaz v kontekstnem meniju
<100/100/100% Orphan segments >
3) Context Menu Command
Ukaz v kontekstnem meniju
<100/100/100% C:\Users\smo\Documents\My Dropbox\verbatim\tm\project_save from verbatim.tmx >
4) Main menu command
Ukaz v glavnem meniju
<66/66/80% >
Dictionary
Multiple Translations
Notes
Comments
Machine Translation
Project autosaved on 12:14 PM
3480/3480 (6113/6780, 7563)
20/25

The match viewer shows the most similar segments from translation memories, both from internal project translation memory created in real time as you translate your project and from ancillary translation memories you have imported from your earlier jobs, or received from your client or translation agency.

When you move to the next segment, the first fuzzy match (the one with the best matching percentage) is automatically selected. You may select a different match by pressing Ctrl+2, 3, 4, or 5. Of course, pressing **Ctrl**+**5** will have no effect, if there is no match #5. To use the selected match in your translation, use Ctrl+R to replace the target field with the match or use Ctrl+I to insert it at the cursor position.

The matching percentage is roughly equivalent to taking the number of common words in the matched and the matching segment and dividing by the number of words in the longer

of the two. The selected fuzzy match is highlighted in bold, words that are missing in the segment you are translating are colored blue and words adjacent to the missing parts green. In the above example the source segment is **Context menu command**. The top match is 100%, because all words match. So do the next two matches, and the match #4 is similar, but different. The line with the matching percentage also includes the name of the translation memory containing the match. If there's no file name displayed, the source is the internal project translation memory. Orphan segments (the match #2) describe segments in the default project translation memory that have no corresponding source segment.

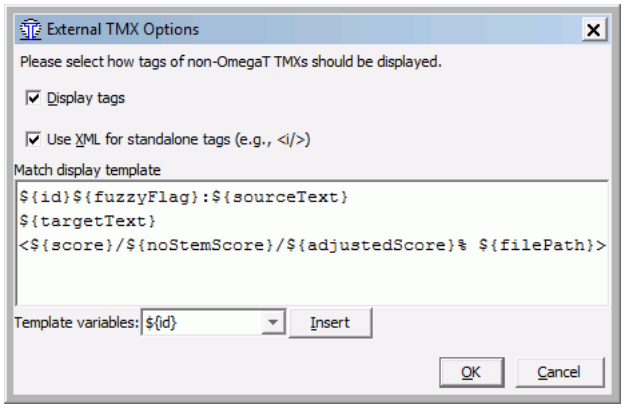
There are actually three match estimates available (66/66/30 in the case of the match #4 above). They are defined as follows:

- match percentage when the tokenizer plugin is used
- default OmegaT match - number of matched words - with numerals and tags ignored - divided by the total word count
- OmegaT match, including numbers, tags

2.2.1. Customizing the Fuzzy matches pane

In Options > External TMXs, a number of variables allow to configure the display of the match pane:

Figure 4.3. Matches pane setup



The figure above shows the default match display template. The contents can be customized using following variables:

Table 4.6. Match pane setup

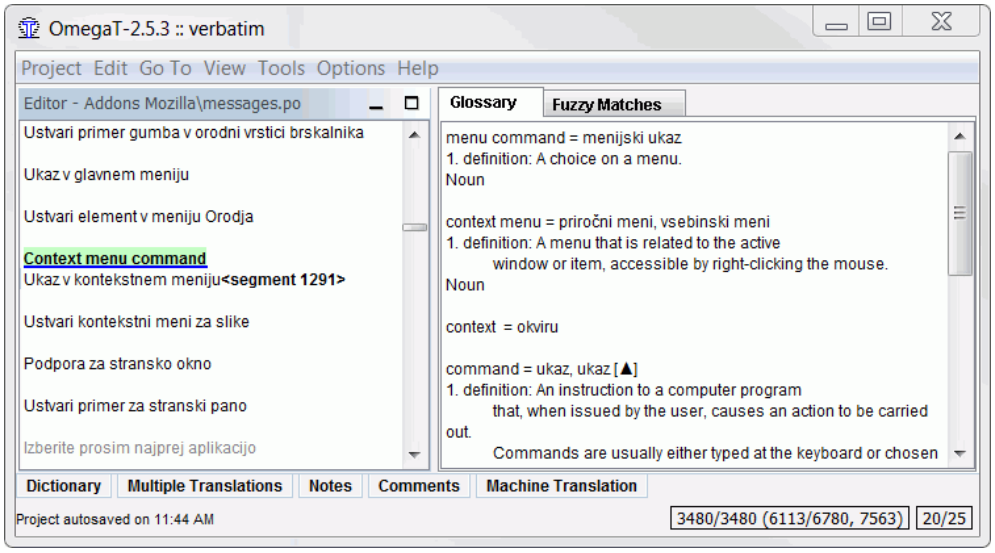
${id}	Number of the match from 1 to 5
${sourceText}	Source text of the match
${targetText}	Target text of the match
${diff}	String showing the differences between the source and the match. *Hint:* use this if the text that you are translating has been updated.
${score}	Percentage with tokenizer
${noStemScore}	Percentage without numbers and tags
${adjustedScore}	Percentage adjusted
${fileNameOnly}	Name of the TMX
${filePath}	Full path of the TMX
${fileShortPath}	Path of the TMX starting from the root of /tm

${creationID}	Author of the match
${creationDate}	Date of the match
${fuzzyFlag}	Indicate that this match is fuzzy (currently only for translations from PO files with the #fuzzy mark)

2.3. Glossary pane

The Glossary pane allows you to access your own collection of expressions and specialist terminology which you have built up in your glossary files. It shows translation of terms found in the current segment. The source segment in the example below was "*Context menu command*", as in the Fuzzy Matches example above, and the terms shown were found in the glossaries, available (Microsoft's Term collection and Slovenian Linux User group Glossary).

Figure 4.4. multi-word entry in the glossary


OmegaT-2.5.3 :: verbatim
Project Edit Go To View Tools Options Help
Editor - Addons Mozilla\messages.po
Ustvari primer gumba v orodni vrstici brskalnika
Ukaz v glavnem meniju
Ustvari element v meniju Orodja
Context menu command
Ukaz v kontekstnem meniju<segment 1291>
Ustvari kontekstni meni za slike
Podpora za stransko okno
Ustvari primer za stranski pano
Izberite prosim najprej aplikacijo
Glossary
Fuzzy Matches
menu command = menijski ukaz
1. definition: A choice on a menu.
Noun
context menu = priročni meni, vsebinski meni
1. definition: A menu that is related to the active
window or item, accessible by right-clicking the mouse.
Noun
context = okviru
command = ukaz, ukaz [▲]
1. definition: An instruction to a computer program
that, when issued by the user, causes an action to be carried
out.
Commands are usually either typed at the keyboard or chosen
Dictionary
Multiple Translations
Notes
Comments
Machine Translation
Project autosaved on 11:44 AM
3480/3480 (6113/6780, 7563)
20/25

If you have TransTips option activated (Options → TransTips), you can right click on the highlighted word in the source segment to open a pop-up menu with suggested translation, as offered by your glossary. Selecting one of them will insert it at the current cursor position into the target segment. You can also highlight your preferred alternative in the glossary pane and insert it into the target by right clicking on the selection.

2.4. Dictionary pane

Dictionaries are the electronic equivalents of printed dictionaries like Merriam Webster, Duden, Larousse etc., that you may have on your desk. See more about them in the chapter on Dictionaries

2.5. Multiple Translations pane

A given source segment may require several different translations, depending on the context. If the current translation of the segment does not fit, the user can select Edit → Create Alternative Translation. The target text entered after that will be treated as an alternative translation of the source segment. You can define one of the alternative - for instance the most probable among them - as default translation by selecting Edit → Use as Default Translation

2.6. Notes pane

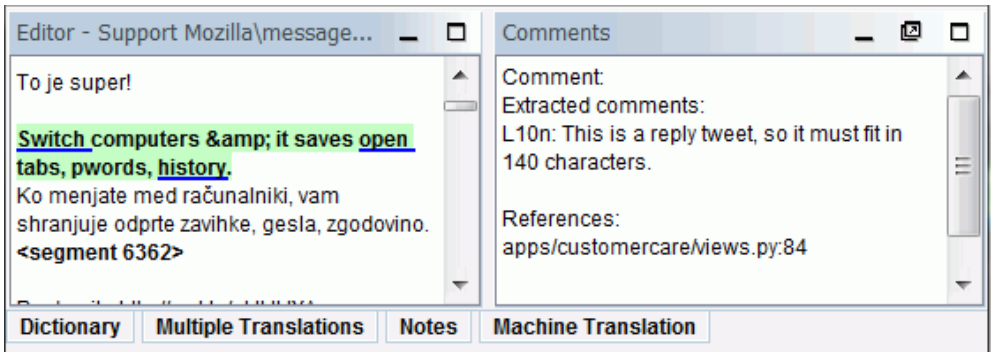
The translator can add notes to the opened segment, for instance to come back later to the segment and redo the translation, check that alternative translations are correct or to ask

colleagues for their opinion. You can browse through notes using GoTo → Next Note and GoTo → Previous Note.

2.7. Comments pane

Some of the file formats, specialized for translation work, for instance PO, allow the inclusion of comments. This way the translator can be provided the context about the segment to be translated. In the example below, the author of the PO file included a warning for the translator to check the length of the translation:

Figure 4.5. Comments pane


Editor - Support Mozilla\message...
To je super!
Switch computers & it saves open tabs, pwords, history.
Ko menjate med računalniki, vam shranjuje odprte zavihke, gesla, zgodovino.
<segment 6362>
Comments
Comment:
Extracted comments:
L10n: This is a reply tweet, so it must fit in 140 characters.
References:
apps/customercare/views.py:84
Dictionary
Multiple Translations
Notes
Machine Translation

2.8. Machine Translation pane

The machine translation pane, when opened, contains the suggestions by machine translation tools for the current segment. Press Ctrl+M to replace the translation of the current segment with the suggested translation. More in the chapter Machine translation

2.9. Main menu

The main menu provides access to all OmegaT functions. See the *Main Menu* appendix for a full description of all menus and menu items. The most frequently used functions are accessible with keyboard shortcuts, so once you become accustomed to them, you will no longer need to browse through the menus while translating. See chapter Menu and Keyboard shortcuts for details.

2.10. Status bar

The status bar displays work-flow related messages at the bottom of the main window. This bar gives the user feedback on specific operations that are in progress. It also displays the number of fuzzy and glossary matches for the current segment.

3. Other windows

3.1. Project files

The Project Files window lists the project files and displays other project information. It is displayed automatically when OmegaT loads a project. Use Ctrl+L to open and **Esc** to close it. The Project Files Window displays the following information:

- the list of all translatable files in the project. These are the files present in the Source files folder in a format that OmegaT is able to recognize. Clicking on any file will open it for translation
- the file currently available in the Editor pane is highlighted with a blue background. Pressing **Enter** will move the Editor pane to the top of the file selected

- File entries include their names, file filter types, their encoding and the number of segments each file contains

- the total number of segments, the number of unique segments in the whole project, and the number of unique segments already translated are shown at the bottom

The set of **Unique** segments is computed by taking all the segments and removing all duplicate segments. (The definition of “unique” is case-sensitive: "Run" and "run" are treated as being different)

The difference between "Number of segments" and "Number of unique segments" provides an approximate idea of the number of repetitions in the text. Note however that the numbers do not indicate how relevant the repetitions are: they could mean relatively long sentences repeated a number of times (in which case you are fortunate) or it could describe a table of keywords (not so fortunate). The project_stats.txt located in the omegat folder of your project contains more detailed segment information, broken down by file.

Modifying the segmentation rules may have the effect of modifying the number of segments/ unique segments. This, however, should generally be avoided once you have started translating the project. See the chapter *Segmentation rules* for more information.

Adding files to the project: You can add source files to the project by clicking on the "Import Source Files..." button. This copies the selected files to the source folder and reloads the project to import the new files. You can also add source files from Internet pages, written in MediaWiki, by clicking on "Import from MediaWiki" button and providing the corresponding URL.

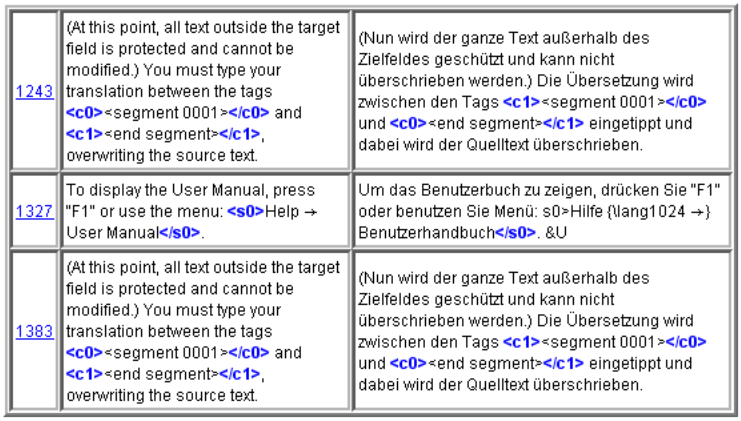
3.2. Search window

You can use the search window to find specific segments in a project. You can also have several search windows open simultaneously. To open a new search window, use **Ctrl**+**F** in the Main window. The search window consists of a text field for search strings or keywords, flags and radio buttons for setting up the search and a display area containing the results of the search. See the chapter *Searches* for more information about the search window.

3.3. Tag validation

The tag validation window detects and lists any tag errors and inconsistencies in the translation. Open the window with Ctrl+T. The window features a 3 column table with a link to the segment and its source and target contents:

Figure 4.6. Tag validation window


1243
(At this point, all text outside the target field is protected and cannot be modified.) You must type your translation between the tags <c0><segment 0001></c0> and <c1><end segment></c1>, overwriting the source text.
(Nun wird der ganze Text außerhalb des Zielfeldes geschützt und kann nicht überschrieben werden.) Die Übersetzung wird zwischen den Tags <c1><segment 0001></c0> und <c0><end segment></c1> eingetippt und dabei wird der Quelltext überschrieben.
1327
To display the User Manual, press "F1" or use the menu: <s0>Help → User Manual</s0>.
Um das Benutzerbuch zu zeigen, drücken Sie "F1" oder benutzen Sie Menü: s0>Hilfe {\lang1024 →} Benutzerhandbuch</s0>. &U
1383
(At this point, all text outside the target field is protected and cannot be modified.) You must type your translation between the tags <c0><segment 0001></c0> and <c1><end segment></c1>, overwriting the source text.
(Nun wird der ganze Text außerhalb des Zielfeldes geschützt und kann nicht überschrieben werden.) Die Übersetzung wird zwischen den Tags <c1><segment 0001></c0> und <c0><end segment></c1> eingetippt und dabei wird der Quelltext überschrieben.

Tags are highlighted in bold blue for easy comparison between the original and the translated contents. Click on the link to jump to the segment in the Editor pane. Correct the error if

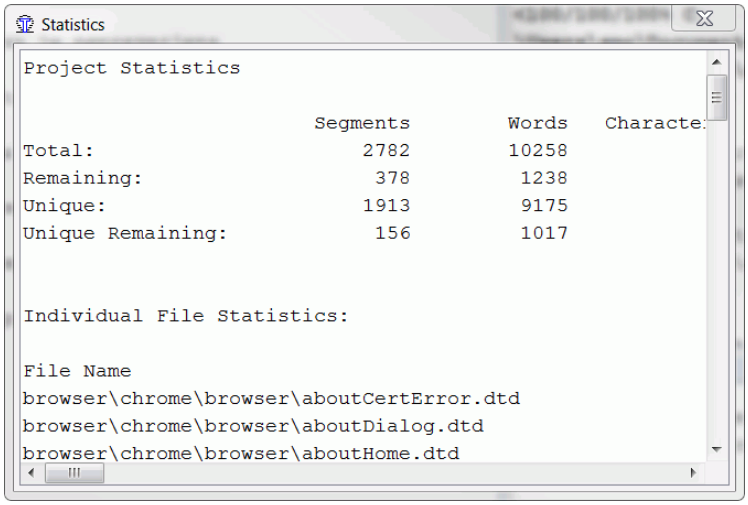
necessary and press Ctrl+T to return to the tag validation window to correct other errors. In the first and third case above tags are paired incorrectly, and in the second case the < sign is missing from the starting tag.

Tag errors are cases in which the tags in the translation do not correspond in order and number to the original segment. Some tag scenarios flagged in the tag validation window are necessary and are benign, others will cause problems when the translated document is created. Tags generally represent some kind of formatting in the original text. Simplifying the original text formatting in the source file before commencing translation greatly contributes to reducing the number of tags.

3.4. Statistics

The statistics window - accessed via **Tools>Statistics** - shows the statistics of the current OmegaT project, both in the summary form as well as in detail for every file to be translated. The statistics shown is available as a tab-separated project_stats.txt file (subfolder omegat), ready to be loaded into a spreadsheet program for the user's convenience. You can use Ctrl +A , Ctrl+C , Ctrl+V to copy/paste the contents.

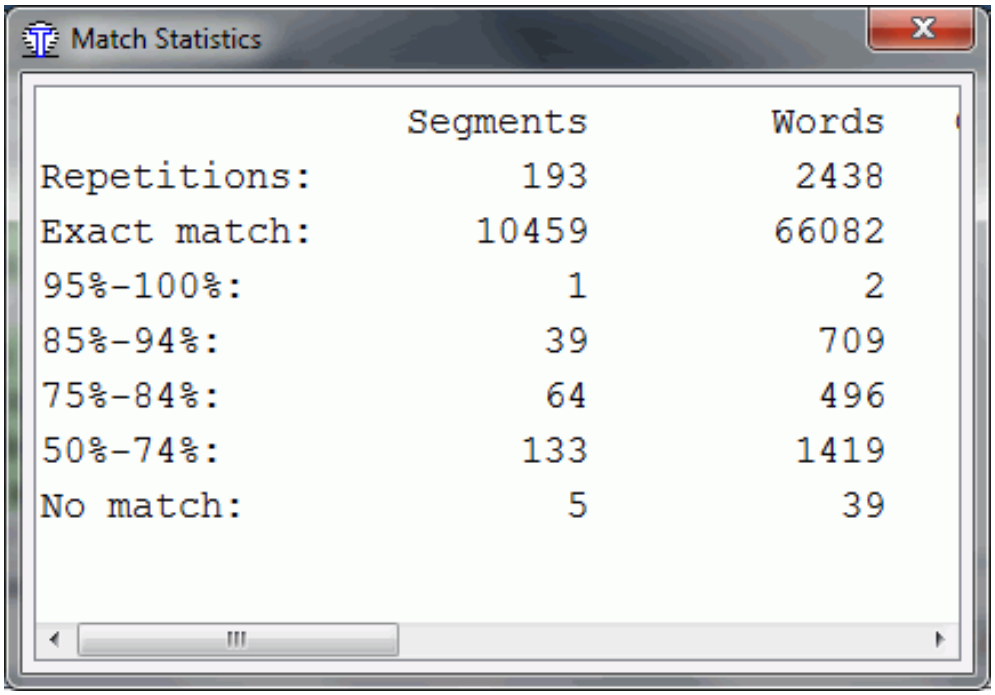
Figure 4.7. project statistics



3.5. Match statistics

The match statistics are accessed viaTools>Match Statistics. The evaluation is rather CPU intensive and can be time-consuming, so a progress bar is shown during the calculation. As far as categories are concerned, the de facto industry standard of classifying matches into the following groups is used: Repetitions, Exact match, 95%-100%, 85%-94%, 75%-84%, 50%-74% and No match. This information is computed for segments as well as for words and for characters (without and including spaces). Note that there could be minor differences between the OmegaT counts and the numbers, provided by other CAT tools.

Figure 4.8. Match statistics



Note that these totals are a good (or as good as they can be) approximation of the work involved in the project and thus can serve as a basis for your cost and price calculations.

Spaces between segments are not taken into account in the last column. Repetitions stand for identical segments present several times in the text. The first segment and its contents will be classified as "no match", and the rest of them as a repetition of the first. If the translation for several identical source segments already exists in the translation memory of the project, these segments, together with other, already translated unique segments, will be classified as an "Exact match". The number of unique segments, if needed, is provided in the standard statistics window, regardless of whether they have been translated or not.

The rest of the categories (50-100%) involves untranslated segments with a fuzzy match. Fuzzy matches can come from the /tm folder as well - and not just from the internal translation memory in /omegaT, as is the case for repetitions and exact matches. The only difference with matches from the project_save translation memory is that external TMs cannot give exact matches, only 100%. If one does not wish to use external TMs for counting, one will either have to empty the /tm folder or change the project setup (temporarily) so that the value for /tm points to a different location.

The Match Statistics are tab-separated and you can use Ctrl+A , Ctrl+C , Ctrl+V to copy/paste them, for instance into a spreadsheet or into your cost-accounting application. Once computed, the data also available in omegat/project_stats_match.txt. Note that the file is time-stamped, as the calculation (contrary to the standard statistics) is not instantaneous and can thus quickly become obsolete.

3.6. Help browser

The help browser (which displays this manual) can be opened by pressing **F1** or navigating to Help → User Manual... in the main menu. In the window, the manual and two buttons are displayed: Back and Contents. The user manual is an HTML document with links to different chapters. Clicking on a link as you would do in a web browser brings the desired page to the front.

The user manual is located in the docs subfolder under the OmegaT installation folder, so you may can, for instance, view the English documentation by opening the docs/en/index.html file in your browser. Opening the user manual in this way also enables you to follow external links, as the built-in help browser does not accept external Internet links.

Chapter 5. Menu and Keyboard shortcuts

1. Main Menu

All of OmegaT's functions are available through the menu bar at the top of the Editor window. Most functions are also available via keyboard shortcuts. Shortcuts are activated by pressing **Ctrl** and a letter. Some shortcuts involve other keys. For readability purposes, letters are written in uppercase here. **Ctrl** is used on Windows, UNIX and UNIX-like operating systems with keyboards featuring a **Ctrl** or **Control** key. Mac users should instead use **Cmd**+**key** instead. The "Cmd" key either has a "command" label or an apple symbol on Apple keyboards.

You can customize existing shortcuts or add new ones according to your needs. See Appendix - Shortcuts Customization

Table 5.1. Main Menu

Project	Edit	Go to	View	Tools	Options	Help

1.1. Project

Table 5.2. Project menu

New		Creates and opens a new project. The dialog to create a project is the same as to edit the project. See Chapter 6, *Project properties*
Open	**Ctrl**+**O**	Opens a previously created project.
Import Source Files...		Copies the selected files to the source folder and reloads the project to load the new files.
Import From MediaWiki...		Imports units from MediaWiki pages, based on the URL entered.
Reload	**F5**	Reloads the project to take external changes in source files, legacy translation memories, glossaries and project settings into account.
Close	**Ctrl**+**Shift**+**W**	Saves the translation and closes the project.
Save	**Ctrl**+**S**	Saves the internal translation memory to the hard disk. OmegaT automatically saves translations every 10 minutes as well as when you close the project or quit OmegaT.
Create translated documents	**Ctrl**+**D**	Creates the target documents based on your translation of the documents' text. The created target documents are located in the target folder.

Properties...	**Ctrl**+**E**	Displays Project properties dialog to edit project languages and folder locations.
Project Files...	**Ctrl**+**L**	Closes or opens the Project files window (depending on whether it is open or closed).
Quit	**Ctrl**+**Q**	Saves the project and quits OmegaT. If you haven't yet saved the project, this manually confirms whether you really wish to quit.

1.2. Edit

Note: Items that are found in most applications (copy/cut/paste) are not displayed in this menu, but are available using your system shortcuts. For example:

Table 5.3. Copy/cut/paste shortcuts

Copy	**Ctrl**+**C**	Copies the selected text to the clipboard.
Cut	**Ctrl**+**X**	Copies the selected text to the clipboard and deletes the selected text.
Paste	**Ctrl**+**V**	Pastes the text from the clipboard at the cursor position.

The Edit menu itself contains the following items:

Table 5.4. Edit menu

Undo Last Action	**Ctrl**+**Z**	Restores the status before the last editing action was taken. This command does not work once the modified segment has been validated.
Redo Last Action	**Ctrl**+**Y**	Restores the status before the last editing action was cancelled. This command does not work once the modified segment has been validated.
Replace With Match	**Ctrl**+**R**	Replaces the whole target segment with the currently selected fuzzy match (by default the first match is selected).
Insert Match	**Ctrl**+**I**	Inserts the currently selected fuzzy match at the cursor position. If part of the target segment has been selected, this function overwrites the selected portion.
Replace with Machine Translation	**Ctrl**+**M**	Replaces the target segment with the translation, provided

		by the selected Machine Translation service. No action is taken, if no Machine Translation service has been activated (see Menu > Options below).
Replace With Source	**Shift**+**Ctrl**+**R**	Replaces the whole target segment with the source.
Insert Source	**Shift**+**Ctrl**+**I**	Inserts the source at the cursor position.
Insert Source Tags	**Shift**+**Ctrl**+**T**	Inserts the source tags at the cursor position. All the relevant tags in the source - i.e. OmegaT as well as custom, Java, printf etc tags - will be inserted into the current target, regardless of whether they are already there or not. Texts, bracketed by tags, are replaced by spaces:. <f0><f1>This is text</f0></f1> thus will be copied as <f0><f1>one_space</f0></f1>
Export Selection	**Shift**+**Ctrl**+**C**	Exports the current selection to a text file for processing. If no text has been selected, the current source segment is written to this file. When the user exits OmegaT, this file is not emptied, in order to be consistent with usual clipboard behavior. The exported contents are copied to the file selection.txt located in the User's preference files folder (see Chapter 8, *OmegaT Files and Folders*).
Create Glossary Entry	**Shift**+**Ctrl**+**G**	Allows the user create an entry in the default glossary file.
Search Project...	**Ctrl**+**F**	Opens a new Search window.
Switch case to...	**Shift**+**F3** (see text)	Changes the case of the highlighted text in the target segment to the selected option (Lower case, Upper case or Title case). Use Shift +F3 to cycle through the three alternatives. If no text is selected, OmegaT selects the word that contains the letter immediately to the right of the cursor.
Select Fuzzy Match #N	**Ctrl**+**#N**	(#N is a digit from 1 to 5) - Selects the Nth fuzzy match displayed in the match

		viewer to replace or insert it to the segment. The Section 2.2, “Fuzzy matches pane” describes the color coding in detail.
Use as Default Translation		If there's several alternative translations available for the active segment, you can label the alternative selected as the default translation. The entry will be greyed, if there's just one translation available.
Create Alternative Translation		The one and the same segment may, depending on the context, require different translations. Select this menu item, if the current translation does not apply and enter the alternative translation.

1.3. Go to

Table 5.5. Go To menu

Next Untranslated Segment	**Ctrl**+**U**	Moves to the next segment that has no equivalent in the translation memory.
Next Segment	**Ctrl**+**N** or **Enter** or **Tab**	Moves to the next segment. If the current segment is the last segment in a file, it moves to the first segment of the next file.
Previous Segment	**Ctrl**+**P** or **Ctrl**+**Enter** or **Ctrl**+**Tab**	Moves to the previous segment. If the current segment is the first one in a file, it moves to the last segment of the previous file.
Segment number...	**Ctrl**+**J**	The segment is opened when its segment number is entered.
Next Note		The next segment with a note attached to it, will open.
Previous Note		The previous segment with a note will open.
Forward in history...	**Ctrl**+**Shift**+**N**	OmegaT remembers the segments executed. With this command you can step forward to the segment, you have previously left by the Back in history...command.
Back in history...	**Ctrl**+**Shift**+**P**	With this command you can move backward one segment at a time, to return later to the current segment using Forward in history... command below.

1.4. View

Table 5.6. View menu

Mark Translated Segments	If checked, the translated segments will be marked in yellow.
Mark Untranslated Segments	If checked, the untranslated segments will be marked in violet.
Display Source Segments	If checked, source segments will be shown and marked in green. If not checked, source segments will not be shown.
Mark Non-Unique Segments	If checked, non-unique segments will be marked in pale grey.
Mark Segments with Notes	If checked, segments with notes will be marked in cyan. This marking has priority over Mark Translated Segments and Mark Untranslated Segments.
Mark Non-breakable Spaces	If checked, non-breakable spaces in the segment will be displayed with a grey background.
Modification Info	Setting the Display Modification option to *Selected* will display the time and the author of the last change in the current segment. Setting it to *All* shows this information for all segments and *None* turns this option off.

1.5. Tools

Table 5.7. Tools menu

Validate Tags	Ctrl+T: Checks for missing or displaced tags in formatted files. Will display a list of segments with tag errors and possible inconsistencies. See Tag Validation and Chapter 12, *Working with formatted text*.
Remove Tags	When enabled, all the formatting tags are removed from source segments. This is useful when dealing with texts where inline formatting has no real use (e.g., OCRed PDF, bad converted .odt or .docx, etc.) Only inline tags are removed, so normally there should be no problem with opening the target documents. Non-visible formatting (i.e., which doesn't appear as tags in the OmegaT editor) is kept in the target document.
Statistics	Opens a new window and displays the project statistics, i.e. the project totals and totals for every file in the project.
Match Statistics	Displays the Match Statistics for the project: the number of repetitions, exact matches, fuzzy matches and no-matches, for segments, words and in characters.

1.6. Options

Table 5.8. Options menu

Use TAB To Advance	Sets segment validation key to Tab instead of the default Enter. This option is useful for some Chinese, Japanese or Korean character input systems.
Always Confirm Quit	The program will see confirmation before closing down.
Machine Translate	Allows you to activate/deactivate the Machine Translation tools offered. When active, **Ctrl**+**M** will insert the suggestion into the target part of the current segment.
Glossary	reserved for future use by plugins
TransTips	Allows you to activate/deactivate *TransTips* feature and set its option *Exact Match.* With *TransTips* activated a right click on a highlighted word in the source will open a pop up menu with the glossary entries for the word you clicked. You can then click on the preferred translation to insert it into the target segment at the current position. With *TransTips/Exact Match* checked, only complete words will be checked, otherwise parts of words will be matched as well.
Font...	Shows the dialog to modify the text display font. Users of old computers who feel window resizing is very slow can try changing the font. See font settings in Miscellanea
File Filters...	Displays the File filters dialog to configure file handling and parsing.
Segmentation...	Opens the Source segmentation dialog to configure text segmentation.
Spell checking...	Displays the Spell checker setup window to install, configure and activate the spell checker.
Editing Behavior...	Displays the Translation editing dialog to configure.
Tag Validation...	For programmers: Allows you to configure the Tag Validator options to check also programming (%...) variables.
Team...	Enter your name here and it will be attached to all segments translated by you.
External TMXs...	Allows the user decide, how tags in foreign TMX files (i.e. not generated by OmegaT) are to be treated.
View...	Contains the option to have all source segments in bold and to mark the first non-unique segment.
Save...	Allows the user select the interval - in minutes and seconds - between consecutive automatic saves of the project. The minimum is 10 seconds.

Proxy login...	Enter your user name and your password, if you use a proxy to access your projects.
Restore Main Window	Restores the components of the main OmegaT window to their default state. Use this feature when you have undocked, moved, or hidden one or more components and you are unable to restore the desired arrangement. It can also be used when panes do not appear as expected following an OmegaT upgrade.

1.7. Help

Table 5.9. Help menu

User Manual...	**F1:** Opens Help browser displaying this manual in a separate window.
About...	Displays copyright, credits and license information.

2. Keyboard shortcuts

The following shortcuts are available from the main window. When another window is on the foreground, click on the main window to bring it to the foreground or press **Esc** to close the other window.

Shortcuts are activated by pressing **Ctrl** and a letter. Some shortcuts involve other keys. For readability purposes, letters are written in uppercase here.

Ctrl is used on Windows, UNIX and UNIX-like operating systems with keyboards featuring a **Ctrl** / **Control key**. Mac users should instead use the cmd+key. On Apple keyboards the cmd key either has a command label or an Apple icon on it.

- Project management
- Editing
- Moving around
- Reference windows
- Other

2.1. Project management

Table 5.10. Project management shortcuts

Open project	**Ctrl**+**O**	Displays a dialog to locate an existing project.
Save	**Ctrl**+**S**	Saves the current work to the internal translation memory (file project_save.tmx located in the project's omegat folder).
Close Project	**Shift**+**Ctrl**+**W**	Closes the current project.
Create Translated Documents	**Ctrl**+**D**	Creates the translated documents in the project's

		Target folder and creates translation memory files (level1, level2 and omegat tmx files) in the project's root folder.
Project properties	**Ctrl**+**E**	Displays the project's settings for modification, if required.

2.2. Editing

Table 5.11. Editing shortcuts

Undo last action	**Ctrl**+**Z**	Undoes the last editing actions in the current target segment
Redo last action	**Ctrl**+**Y**	Redoes the last editing actions in the current target segment
Select match #N	**Ctrl**+**#N**	#N is a digit from 1 to 5. The shortcut selects the Nth match displayed in the match window (the first match is selected by default)
Replace with match	**Ctrl**+**R**	Replaces the current target segment contents with the selected match (the first match is selected by default)
Insert match	**Ctrl**+**I**	Inserts the selected match at the cursor position in the current target segment (the first match is inserted by default)
Replace with source	**Ctrl**+**Shift**+**R**	Replaces the current target segment contents with the source text contents
Insert source	**Ctrl**+**Shift**+**I**	Inserts the source text contents into the target segment at the cursor position
Insert Source Tags	**Ctrl**+**Shift**+**T**	Inserts the source tags into the target segment at the cursor position
Search project	**Ctrl**+**F**	Displays a dialog to conduct searches in the project
Replace with Machine Translation	**Ctrl**+**M**	Replaces the target segment with the machine translation of the source. No action, if machine tools are deactivated (see Menu > Options > Machine Translate)
Export Selection	**Shift**+**Ctrl**+**C**	Exports the current selection to a text file for processing.
Create Glossary Entry	**Shift**+**Ctrl**+**G**	Allows the user create an entry in the default glossary file.

2.3. Moving around

Table 5.12. Moving around shortcuts

Next Untranslated Segment	**Ctrl**+**U**	Moves the editing field to the next segment that is not registered in the project's translation memory
Next Segment	**Ctrl**+**N**, **Enter** or **Return**	Moves the editing field to the next segment.
Previous Segment	**Ctrl**+**P**	Moves the editing field to the previous segment
Segment number...	**Ctrl**+**J**	Moves to the segment number entered
Back in history...	**Ctrl**+**Shift**+**P**	Moves one segment back in history
Forward in history...	**Ctrl**+**Shift**+**N**	Moves one segment forward in history

2.4. Other

Table 5.13. Various shortcuts

Project files listing	**Ctrl**+**L**	Displays the Project files listing
Validate Tags	**Ctrl**+**T**	Opens the Tag validation window.
Export Selection	**Shift**+**Ctrl**+**C**	Exports the current selection or the current source, if no text has been selected. The text is exported to a plain text file.
Search Project	**Ctrl**+**F**	Opens a new Search window.
Help files	**F1**	Displays the OmegaT help files in a separate window

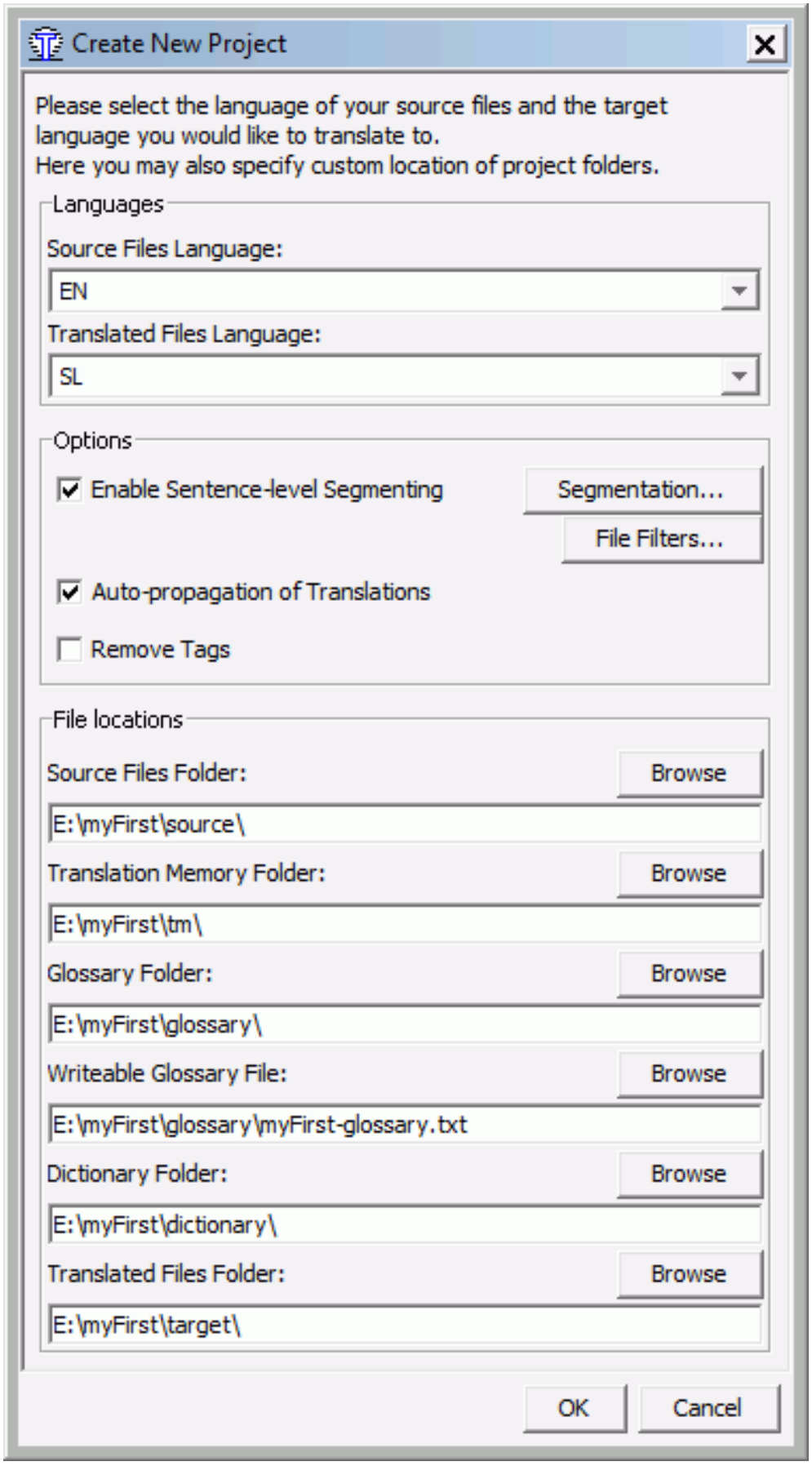
Chapter 6. Project properties

1. Properties dialog

The Project → Properties... (**Ctrl+E**) dialog is used to define and modify the project folders and languages.

It is possible to modify the project properties during a translation session. Note that changes to the project setup may have some consequences, especially, when the project has already been started. Until you have some experience with OmegaT, it is safest to consider all settings final once the translation has started – unless of course you realize a major mistake has been made. See the section Preventing data loss for ways and means of protecting your work.

Figure 6.1. Project properties window



It contains:

- **An area for setting project source and target languages**

  You can either enter the source and target languages by hand or use the drop down menus. Bear in mind that changing the languages may render the currently used translation memories useless since their language pair may not longer match the new languages.

- **Enable Sentence-level segmentation** check box

  The segmentation settings only address the way the source files are handled by OmegaT. The predominant way of segmenting the sources is the sentence-level segmenting, so this check box should in a normal case be left checked.

  In some seldom cases the alternative, i.e. segmenting by paragraphs, may be preferred. Changing this flag does not modify the segmentation of already existing translation memories. If you decide mid-translation to switch from sentence to paragraph translation, the internal translation memory of the project will not be changed (OmegaT may upgrade old translation memories that did not use sentence segmentation, but not vice versa).

  Changing segmentation settings may cause some already translated segments to be split or merged. This will effectively return them to the "untranslated" status, as they will no longer match segments recorded in the project memory, even though their original translation is still there.

- **Segmentation...**button

  The segmentation rules are generally valid across all the projects. The user, however, may need to generate a set of rules, specific to the project in question. Use this button to open a dialog,activate the check box Project specific segmentation rules, then proceed to adjust the segmentation rules as desired. The new set of rules will be stored together with the project and will not interfere with the general set of segmentation rules. To delete project specific segmentation rules, uncheck the check box. See chapter Source Segmentation for more information on segmentation rules.

  Hint: the set of segmentation rules for a given project is stored as project/omegat/segmentation.conf.

- **File Filters...**button

  In a similar fashion as above the user can create project-specific File filters, which will be stored together with the project and will be valid for the current project only. To create a project-specific set of file filters, click on the File filter ... button, then activate Enable project specific filters check box in the window, that opens. A copy of the changed filters configuration will be stored with the project. To delete project specific file filters, uncheck the check box. Note that in the menu Options->File Filters, the global user filters are changed, not the project filters. See chapter File filtersfor more on the subject.

  Hint: the set of file filters for a given project is stored as project/omegat/filters.xml.

- **Autopropagation**check box

  In case there are non-unique segments in source documents, the Auto-propagation check box offers the user the following two possibilities as regards automatic translation: if checked, the first translated segment will be assumed as the default translation and its target text will be automatically used for later hits during the translation process. Mistranslated segments can of course be corrected later manually using Create Alternative Translation. If the Auto-propagation check box is not checked, the segments with alternative translations are left untranslated until the user has decided which translation is to be used.

- **Remove tags** check box

When enabled, all the formatting tags are removed from source segments. This is especially useful when dealing with texts where inline formatting is not really useful (e.g., OCRed PDF, bad converted .odt or .docx, etc.) In a normal case it should always be possible to open the target documents, as only inline tags are removed. Non-visible formatting (i.e., which doesn't appear as tags in the OmegaT editor) is retained in target documents.

- **An area for setting project folders**

  Here you can select different subfolders, for instance the subfolder with source files, subfolder for target files etc. If you enter names of folders that do not exist yet, OmegaT creates them for you. In case you decide to modify project folders, keep in mind that this will not move existing files from old folders to the new location.

Chapter 7. File Filters

OmegaT features highly customizable filters, enabling you to configure numerous aspects. File filters are pieces of code capable of:

- Reading the document in some specific file format. For instance, plain text files.
- Extracting the translatable content out of the file.
- Automating modifications of the translated document file names by replacing translatable contents with its translation.

To see which file formats can be handled by OmegaT, see the menu Options > File Filters ...

Most users will find the default file filter options sufficient. If this is not the case, open the main dialog by selecting **Options → File Filters...** from the main menu. You can also enable project-specific file filters, which will only be used on the current project, by selecting the **File Filters...** option in Project Properties.

You can enable project specific filters via the **Project → Properties...**. Click on File Filters button and activate the check box Enable project specific filters. A copy of the filters configuration will be stored with the project in this case. If you later change filters, only the project filters will be updated, while the user filters stay unchanged.

Warning! Should you change filter options whilst a project is open, you must reload the project in order for the changes to take effect.

1. File filters dialog

This dialog lists available file filters. Should you wish not to use OmegaT to translate files of a certain type, you can turn off the corresponding filter by deactivating the check box beside its name. OmegaT will then omit the appropriate files while loading projects, and will copy them unmodified when creating target documents. When you wish to use the filter again, just tick the check box. Click **Defaults** to reset the file filters to the default settings. To edit which files in which encodings the filter is to process, select the filter from the list and click **Edit.**

2. Filter options

Several filters (Text files, XHTML files, HTML and XHTML files, OpenDocument files and Microsoft Open XML files) have one or more specific options. To modify the options select the filter from the list and click on **Options**. The available options are:

Text files

- *Paragraph segmentation on line breaks, empty lines or never:*

  if sentence segmentation rules are active, the text will further be segmented according to the option selected here.

PO files

- *Allow blank translations in the target file*:

  If on, when a PO segment (which may be a whole paragraph) is not translated, the translation will be empty in the target file. Technically speaking, the msgstr segment in the PO target file, if created, will be left empty. As this is the standard behavior for PO files, it is on by default. If the option is off, the source text will be copied to the target segment.

- *Skip PO header*

PO header will be skipped and left unchanged, if this option is checked.

- *Auto replace 'nplurals=INTEGER; plural=EXPRESSION;' in header*

  The option allows OmegaT to override the specification in the PO file header and use the default for the selected target language.

XHTML Files

- *Add or rewrite encoding declaration in HTML and XHTML files*: frequently the target files must have the encoding character set different from the one in the source file (whether it is explicitly defined or implied). Using this option the translator can specify, whether the target files are to have the encoding declaration included. For instance, if the file filter specifies UTF8 as the encoding scheme for the target files, selecting Always will assure that this information is included in the translated files.
- *Translate the following attributes*: the selected attributes will appear as segments in the Editor window.
- *Start a new paragraph on: the
* HTML tag will constitute a paragraph for segmentation purposes.
- *Skip text matching regular expression*: the text matching the regular expression gets skipped. It is shown rendered red in the tag validator. Text in source segment that matches is shown in italic.
- *Do not translate the content attribute of meta-tags ... :* The following meta-tags will not be translated.
- *Do not translate the content of tags with the following attribute key-value pairs (separate with commas)*: a match in the list of key-value pairs will cause the content of tags to be ignored

  It is sometimes useful to be able make some tags untranslatable based on the value of attributes. For example,   You can define key-value pairs for tags to be left untranslated. For the example above, the field would contain: class=hide, translate=no

Microsoft Office Open XML files

You can select which elements are to be translated. They will appear as separate segments in the translation.

- **Word:** non-visible instruction text, comments, footnotes, endnotes, footers
- **Excel:** comments, sheet names
- **Power Point**: slide comments, slide masters, slide layouts
- **Global:** charts, diagrams, drawings, WordArt
- **Other Options:**
  - *Aggregate tags*: if checked, tags without translatable text between them will be aggregated into single tags.
  - *Preserve spaces for all tags*: if checked, "white space" (i.e., spaces and newlines) will be preserved, even if not set technically in the document

HTML and XHTML files

- *Add or rewrite encoding declaration in HTML and XHTML files*: Always (default), Only if (X)HTML file has a header, Only if (X)HTML file has an encoding declaration, Never

- *Translate the following attributes*: the selected attributes will appear as segments in the Editor window.
- *Start a new paragraph on*: the
 HTML tag will constitute a paragraph for segmentation purposes.
- *Skip text matching regular expression*: The text, matching the regular expression, will be skipped.
- *Do not translate the content attribute of meta-tags ... :* The following meta-tags will not be translated.
- *Do not translate the content of tags with the following attribute key-value pairs (separate with commas)*: a match in the list of key-value pairs will cause the content of tags to be ignored

Text files

- *Paragraph segmentation on line breaks, empty lines or never*:

  if sentence segmentation rules are active, the text will further be segmented according to the option selected here.

Open Document Format (ODF) files

- You can select which of the following items are to be translated:

  index entries, bookmarks, bookmark references, notes, comments, presentation notes, links (URL), sheet names

3. Edit filter dialog

This dialog enables you to set up the source filename patterns of files to be processed by the filter, customize the filenames of translated files, and select which encodings should be used for loading the file and saving its translated counterpart. To modify a file filter pattern, either modify the fields directly or click **Edit.** To add a new file filter pattern, click **Add**. The same dialog is used to add a pattern or to edit a particular pattern. The dialog is useful because it includes a special target filename pattern editor with which you can customize the names of output files.

3.1. Source file type, filename pattern

When OmegaT encounters a file in its source folder, it attempts to select the filter based upon the file's extension. More precisely, OmegaT attempts to match each filter's source filename patterns against the filename. For example, the pattern *.xhtml matches any file with the .xhtml extension. If the appropriate filter is found, the file is assigned to it for processing. For example, by default, XHTML filters are used for processing files with the .xhtml extension. You can change or add filename patterns for files to be handled by each file. Source filename patterns use wild card characters similar to those used in **Searches.** The '*' character matches zero or more characters. The '?' character matches exactly one character. All other characters represent themselves. For example, if you wish the text filter to handle readme files (readme, read.me, and readme.txt) you should use the pattern read*.

3.2. Source and Target file encoding

Only a limited number of file formats specify a mandatory encoding. File formats that do not specify their encoding will use the encoding you set up for the extension that matches their name. For example, by default .txt files will be loaded using the default encoding of your operating system. You may change the source encoding for each different source filename

pattern. Such files may also be written out in any encoding. By default, the translated file encoding is the same as the source file encoding. Source and target encoding fields use combo boxes with all supported encodings included. <auto> leaves the encoding choice to OmegaT. This is how it works:

- OmegaT identifies the source file encoding by using its encoding declaration, if present (HTML files, XML based files)
- OmegaT is instructed to use a mandatory encoding for certain file formats (Java properties etc)
- OmegaT uses the default encoding of the operating system for text files.

3.3. Target filename

Sometimes you may wish to rename the files you translate automatically, for example adding a language code after the file name. The target filename pattern uses a special syntax, so if you wish to edit this field, you must click **Edit...**and use the Edit Pattern Dialog. If you wish to revert to default configuration of the filter, click **Defaults.** You may also modify the name directly in the target filename pattern field of the file filters dialog. The Edit Pattern Dialog offers among others the following options:

- Default is ${filename}– full filename of the source file with extension: in this case the name of the translated file is the same as that of the source file.
- ${nameOnly}– allows you to insert only the name of the source file without the extension.
- ${extension} - the original file extension
- ${targetLocale}– target locale code (of a form "xx_YY").
- ${targetLanguage}– the target language and country code together (of a form "XX-YY").
- ${targetLanguageCode} – the target language - only "XX"
- ${targetCountryCode}– the target country - only "YY"
- ${timestamp-????} – system date time at generation time in various patterns

  See Oracle documentation [http://docs.oracle.com/javase/1.4.2/docs/api/java/text/SimpleDateFormat.html] for examples of the "SimpleDateFormat" patterns
- ${system-os-name} - operating system of the computer used
- ${system-user-name} - system user name
- ${system-host-name} - system host name
- ${file-source-encoding} - source file encoding
- ${file-target-encoding} - target file encoding
- ${targetLocaleLCID} - Microsoft target locale

Additional variants are available for variables ${nameOnly} and ${Extension}. In case the file name has ambivalent name, one can apply variables of the form ${name only-*extension number*} and ${extension-*extension number}* . If for example the original file is named Document.xx.docx, the following variables will give the following results:

- ${nameOnly-0} Document
- ${nameOnly-1} Document.xx
- ${nameOnly-2} Document.xx.docx

- ${extension-0} docx
- ${extension-1} xx.docx
- ${extension-2} Document.xx.docx

Chapter 8. OmegaT Files and Folders

OmegaT works with three types of files.

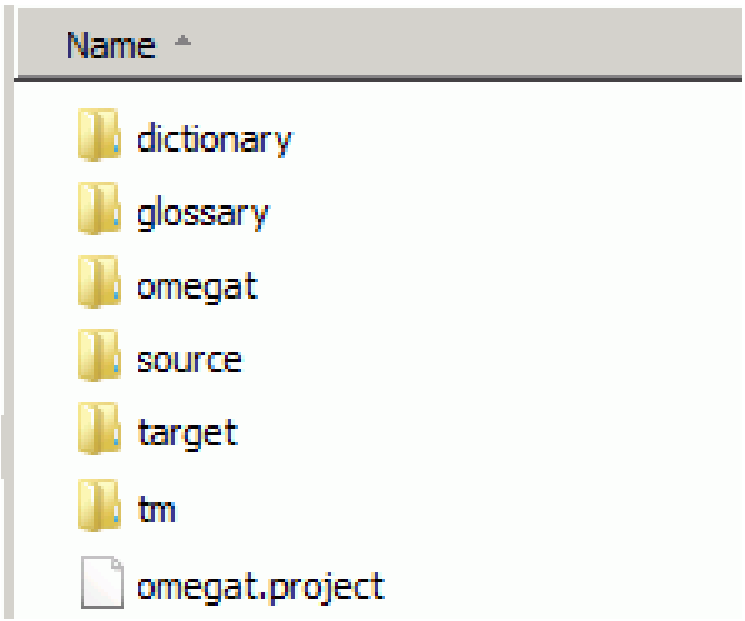
- Translation project files: These constitute a translation project. Losing them may affect the project's integrity and your ability to complete a job. Project files are the most important files in OmegaT. They are the files you deal with on a daily basis while translating.
- User settings files: These are created when OmegaT's behavior is modified by user preference settings. Losing them usually results in OmegaT reverting to its "factory settings". This can sometimes cause a little trouble when you are in the middle of a translation.
- Application files: These are included in the package you download. Most of them are required in order for OmegaT to function properly. If for some reason these files are lost or corrupted, simply download and/or reinstall OmegaT to restore them all.

1. Translation project files

An OmegaT translation project consists of a number of files and folders.

When you create a translation project, OmegaT automatically creates a folder with the specified name, and a list of folders:

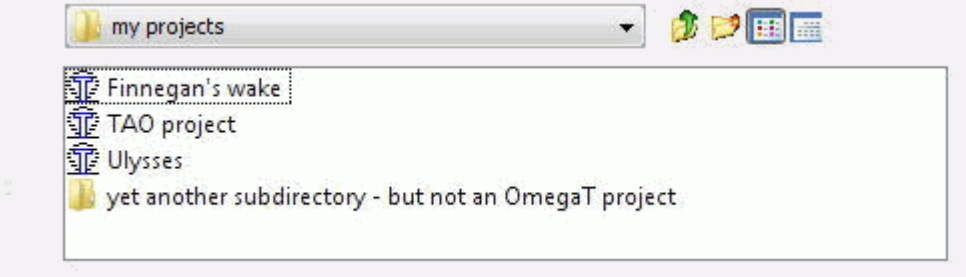
Figure 8.1. OmegaT project



Alternate locations for some of the folders can be chosen at project creation or during the translation. It is therefore possible to select existing folders or create folders in locations that reflect your work flow and project management habits. To change the location of folders after a project has been created, open Project > Properties... in the menu or with Ctrl+E and make the necessary changes.

In a file manager a translation project looks and acts just like any other folder. In the following example the folder my projects contains three OmegaT projects:

Figure 8.2. OmegaT projects and subfolders



Double clicking the item with the OmegaT icon is sufficient to open the project. A translation project **Example_Project** created with the default settings will be created as a new subfolder with the following structure:

1.1. Top folder

Top folder of a project always contains the file OmegaT.Project, containing project parameters as defined in the Project properties window (Project > Properties). While the translation is progressing, additional files (*project_name*-omegat.tmx, project_name-level1.tmx and project_name-level2.tmx) are created (and updated during the process of translation) in this folder. They contain the one and the same translation memory contents in different forms, to be used in future projects.

1.2. Subfolder dictionary

Initially empty, this folder will contain dictionaries you have added to the project. See chapter Dictionaries for more on this subject.

1.3. Subfolder glossary

This folder is initially empty. It will contain glossaries you will be using in the project. See chapter Glossaries for more on this subject.

1.4. Subfolder omegat

The **omegat** subfolder contains at least one and possibly several other files. The most important file here is the project_save.tmx, that is the working translation memory for the project. Backups of this file (with extension bak) are added progressively to this subfolder, first at the beginning of the translation session, at its end, and while the translation progresses. This way an inadvertent data loss is averted - see Preventing Data Loss in chapter Miscellanea.

During translation additional files may get created in this subfolder as follows

stats.txt	contains the current statistics of the current project. You can view it by selecting Tools > Statistics
ignored_words.txt. learned_words.txt	are created and used by the spell checker. If you already have collected words you wish the spell checker to ignore / accept, you just need to copy the corresponding two files into the omegatsubfolder of your current project.
project_stats_match.txt	contains the latest project match statistics, generated by Tools > Match Statistics
segmentation.conf	if existing, it contains project-specific segmentation rules, if requested in Project > Properties ... See Chapter Project properties

filters.xml	if existing, it contains project-specific file filters, if requested in Project > Properties ... See Chapter Project properties

1.5. Subfolder source

The source subfolder contains files to be translated. You can add the files to it later. Note that the structure of the source subfolder may take any form you like. If the files to be translated are parts of a tree structure (as in a website), you need only specify the top-level subfolder and OmegaT will maintain the entire contents, while keeping the tree structure intact.

1.6. Subfolder target

This subfolder is initially empty. To add contents to it, select Project → Create Translated Documents (**Ctrl+D**). Files within the source folder, whether translated or not, are then generated here, with the same hierarchy as present in the source subfolder. The contents of the target subfolder will reflect the current state of the translation, as present in the project translation memory, saved in the current **/omegat/project_save.tmx**. Untranslated segments will hereby remain in the source language.

2. User settings files

User files contain the information, applicable to all the projects for a given user;

logs/OmegaT.log	This file records Java error messages while OmegaT is running. Should OmegaT appear to be behaving erratically, it is important to include this file or the relevant part in any bug report
script/	folder, containing script files for the script plugin, if installed
filters.xml	user's default file filters
omegat.prefs	OmegaT preferences
segmentation.conf	user's default segmentation rules
uiLayout.xml	An xml file with all the GUI accessible option settings

Note that default segmentation rules and file filters can be overridden by project-specific setup (see above). The location of user files depends upon the platform you use:

Windows 2000 and XP	Documents and Settings\<User Name>\Application Data\OmegaT
Windows Vista and 7	Users\<User Name>\AppData\Roaming\OmegaT
Windows other	<Something>\OmegaT (<Something> corresponds to the location of the "home" folder as determined by Java)
Linux/Solaris/FreeBSD	<User Home>/.omegat (.omegat is a folder, the dot preceding its name makes it invisible unless you type ls -a or an equivalent command)
MAC OS X	<User Home>/Library/Preferences/OmegaT
Other	<User Home>

3. Application files

OmegaT is supplied as a package that can be downloaded from SourceForge. Here a platform-independent package in a standard Java form is considered. Alternatives include a Linux .tar package, a Windows installer – with or without a Java Runtime Environment –, a Mac OS X installer, and a source code package for developers.

The platform-independent package can be used on any platform with a working Java 1.5 runtime environment, including the platforms for which a specific package also exists. It is provided as a compressed file (zip or tar archive) that you must extract to the folder of your choice for installation. The file can usually be extracted by double-clicking on the downloaded package. Once the archive has been extracted, a folder containing the following contents is created:

File/ subfolder	**Contents**
/docs/	All the user manual files can be found in this folder. You can open them in an Internet browser to obtain access to external links.
/images/	Icons and logo graphics
/lib/	Contains Java files, necessary to the operation of OmegaT.
join.html	This is an ordinary html file that, when opened in your Internet browser, directs you to the OmegaT user group hosted on Yahoo! Groups. Joining is not necessary, but will provide you with access to additional services, such as files, questionnaires, and the opportunity to take part in OmegaT-related discussions. The group archives are public and can be viewed without subscription to the group.
changes.txt	A relatively detailed list of modifications between this version and the preceding versions.
license.txt	The GNU GENERAL PUBLIC LICENSE. This license allows you to do certain things with OmegaT, including modifying and distributing it. If you are interested in modifying or distributing OmegaT, read this document carefully and ensure you understand its implications before doing anything. If in doubt, don't hesitate to ask project members directly either by sending them an e-mail from the SourceForge page or by sending a public mail to the user group.
doc-license.txt	The GNU GENERAL PUBLIC LICENSE. This license covers the documentation. See above.
readme.txt	This file is very important and you should make sure you read it before launching OmegaT. It includes general information on OmegaT, where to find more information, how to contribute, etc. It has been translated into a number of languages.
OmegaT	A text file containing two lines:

	#!/bin/bash java java -jar OmegaT.jar $* Linux and OS X users may find this file useful. Make it executable (chmod +x OmegaT) from the command line after making sure you are in the OmegaT application folder. You will then be able to launch OmegaT by executing this file from the command line
OmegaT.bat	A batch file, used to launch OmegaT from the Windows command line. It contains just the following line: java -jar OmegaT.jar %*
OmegaT.jar	The main OmegaT application. To launch OmegaT, you must launch this file either from the command line or from your file manager, usually by double-clicking it.

Chapter 9. Files to translate

1. File formats

You can use OmegaT to translate files in a number of file formats. There are basically two types of file formats, plain text and formatted text.

1.1. Plain text files

Plain text files contain text only, so their translation is as simple as typing the translation. There are several methods to specify the file's encoding so that its contents are not garbled when opened in OmegaT. Such files do not contain any formatting information beyond the "white space" used to align text, indicate paragraphs or insert page breaks. They are not able to contain or retain information regarding the color, font etc of the text. Currently, OmegaT supports the following plain text formats:

- ASCII text (.txt, etc.)
- Encoded text (*.UTF8)
- Java resource bundles (*.properties)
- PO files (*.po)
- INI (key=value) files (*.ini)
- DTD files (*.DTD)
- DokuWiki files (*.txt)
- SubRip title files (*.srt)
- Magento CE Locale CSV files (*.csv)

Other plain text file types can be handled by OmegaT by associating their file extension to a supported file type (for example, .pod files can be associated to the ASCII text filter) and by pre-processing them with specific segmentation rules.

PO files can contain both the source and the target text. Seen from this point of view, they are plain text files *plus* translation memories. If for a given source segment there is as yet no existing translation in the project translation memory (project_save.tmx), the current translation will be saved in the project_save.tmx as the default translation. In case, however, the same source segment already exists with a different translation, the new translation will be saved as an alternative.

1.2. Formatted text files

Formatted text files contain information such as font type, size, color etc. as well as text. They are commonly created in word processors or HTML editors. Such file formats are designed to hold formatting information. The formatting information can be as simple as “this is bold”, or as complex as table data with different font sizes, colors, positions, etc. In most translation jobs, it is considered important for the formatting of the original text to be retained in the translation. OmegaT allows you to do this by marking the characters/words that have a special formatting with easy-to-handle tags. Simplifying the original text formatting greatly contributes to reducing the number of tags. Where possible, unifying the fonts, font sizes, colors, etc. used in the document simplifies the task of translation and reduces the possible number of tag errors. Each file type is handled differently in OmegaT. Specific behavior can be set up in the file filters. At the time of writing, OmegaT supports the following formatted text formats:

- ODF - OASIS Open Document Format (*.ods, *.ots, *.odt, *.ott, *.odp, *.otp)
- Microsoft Office Open XML (*.docx, *.dotx, *.xlsx, *.xltx, *.pptx)
- (X)HTML (*.html, *.xhtml,*.xht)
- HTML Help Compiler (*.hhc, *.hhk)
- DocBook (*.xml)
- XLIFF (*.xlf, *.xliff, *.sdlxliff) - of the source=target variety
- QuarkXPress CopyFlowGold (*.tag, *.xtg)
- ResX files (*.resx)
- Android resource (*.xml)
- LaTex (*.tex, *.latex)
- Help (*.xml) and Manual (*.hmxp) files
- Typo3 LocManager (*.xml)
- WiX Localization (*.wxl)
- Iceni Infix (*.xml)
- Flash XML export (*.xml)
- Wordfast TXML (*.txml)
- Camtasia for Windows (*.camproj)
- Visio (*.vxd)

Other formatted text file types may also be handled by OmegaT by associating their file extensions to a supported file type, assuming that the corresponding segmentation rules will segment them correctly.

2. Other file formats

Other plain text or formatted text file formats suitable for processing in OmegaT may also exist.

External tools can be used to convert files to supported formats. The translated files will then need to be converted back to the original format. For example, if you have an outdated Microsoft Word version, that does not handle the ODT format, here's a round trip for Word files with the DOC extension:

- import the file into ODF writer
- save the file in ODT format
- translate it into the target ODT file
- load the target file in ODF writer
- save the file as a DOC file

The quality of formatting of the translated file will depend on the quality of the round-trip conversion. Before proceeding with such conversions, be sure to test all options. Check the OmegaT home page [http://www.omegat.org] for an up-to-date listing of auxiliary translation tools.

3. Right to left languages

Justification of source and target segments depends upon the project languages. By default, left justification is used for Left-To-Right (LTR) languages and right justification for Right-To-Left (RTL) languages. You can toggle between different display modes by pressing **Shift**+**Ctrl**+**O** (this is the letter O and not the numeral 0). The **Shift**+**Ctrl**+**O** toggle has three states:

- default justification, that is as defined by the language
- left justification
- right justification

Using the RTL mode in OmegaT has no influence whatsoever on the display mode of the translated documents created in OmegaT. The display mode of the translated documents must be modified within the application (such as Microsoft Word) commonly used to display or modify them (check the relevant manuals for details). Using **Shift**+**Ctrl**+**O** causes both text input and display in OmegaT to change. It can be used separately for all three panes (Editor, Fuzzy Matches and Glossary) by clicking on the pane and toggling the display mode. It can also be used in all the input fields found in OmegaT - in the search window, for segmentation rules etc.

Mac OS X users, note: use **Shift**+**Ctrl**+**O** shortcut and **not** cmd+Ctrl+O.

3.1. Mixing RTL and LTR strings in segments

When writing purely RTL text, the default (LTR) view may be used. In many cases, however, it is necessary to embed LTR text in RTL text. For example, in OmegaT tags, product names that must be left in the LTR source language, place holders in localization files, and numbers in text. In cases like these it becomes necessary to switch to RTL mode, so that the RTL (in fact bidirectional) text is displayed correctly. It should be noted that when OmegaT is in RTL mode, both source and target are displayed in RTL mode. This means that if the source language is LTR and the target language is RTL, or vice versa, it may be necessary to toggle back and forth between RTL and LTR modes to view the source and enter the target easily in their respective modes.

3.2. OmegaT tags in RTL segments

As stated above, OmegaT tags are LTR strings. When translating between RTL and LTR languages, correctly reading the tags from the source and entering them properly in the target may require the translator to toggle between LTR and RTL modes numerous times.

If the document allows, the translator is strongly encouraged to remove style information from the original document so that as few tags as possible appear in the OmegaT interface. Follow the indications given in Hints for tags management. Frequently validate tags (see Tag validation) and produce translated documents (see below and Menu) at regular intervals to make it easier to catch any problems that arise. A hint: translating a plain text version of the text and adding the necessary style in the relevant application at a later stage may turn out to be less hassle.

3.3. Creating translated RTL documents

When the translated document is created, its display direction will be the same as that of the original document. If the original document was LTR, the display direction of the target document must be changed manually to RTL in its viewing application. Each output format has specific ways of dealing with RTL display; check the relevant application manuals for details.

For .docx files, a number of changes are however done automatically:

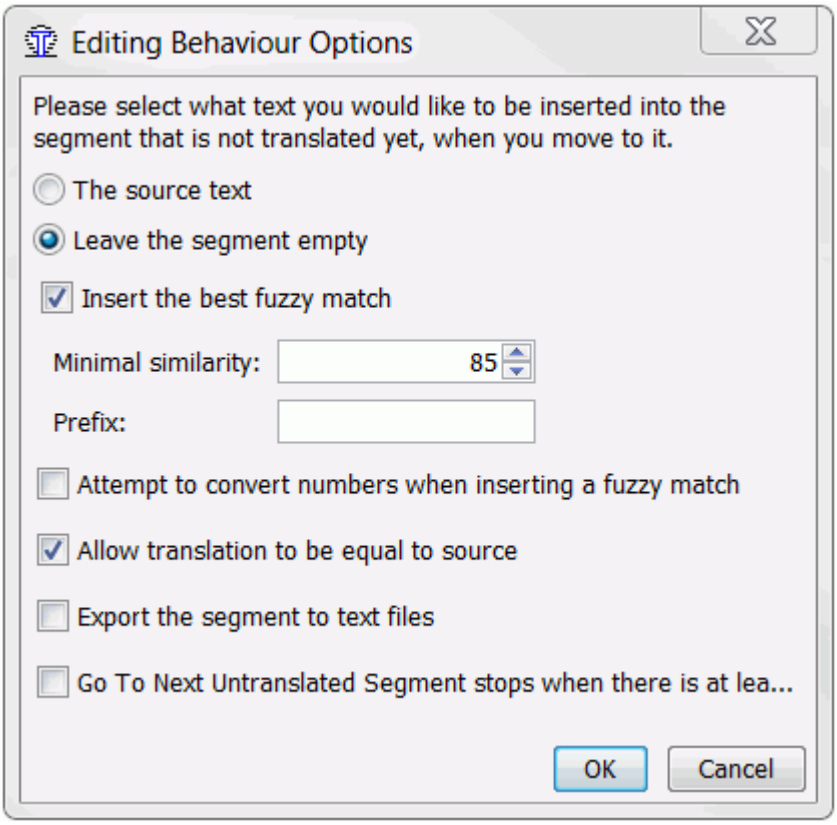
- Paragraphs, sections and tables are set to bidi
- Runs (text elements) are set to RTL

To avoid changing the target files display parameters each time the files are opened, it may be possible to change the source file display parameters such that such parameters are inherited by the target files. Such modifications are possible in ODF files for example.

Chapter 10. Editing behavior

The dialog in Options → Editing Behavior... enables the user to select, how the current segment in the editing field is to be initialized and handled:

Figure 10.1. Editing behavior options


Editing Behaviour Options
Please select what text you would like to be inserted into the segment that is not translated yet, when you move to it.
The source text
Leave the segment empty
Insert the best fuzzy match
Minimal similarity:
85
Prefix:
Attempt to convert numbers when inserting a fuzzy match
Allow translation to be equal to source
Export the segment to text files
Go To Next Untranslated Segment stops when there is at lea...
OK
Cancel

You translate your files by moving from segment to segment, editing each current segment in turn. When moving between segments, you may wish to populate the editing field with an existing translation in the fuzzy match pane or with the source text. In Options → Editing Behavior... OmegaT offers you the following alternatives:

The source text	You can have the source text inserted automatically into the editing field. This is useful for texts containing many trade marks or other proper nouns you which must be left unchanged.
Leave the segment empty	OmegaT leaves the editing field blank. This option allows you to enter the translation without the need to remove the source text, thus saving you two keystrokes (**Ctrl+A** and **Del).** Empty translations are now allowed. They are displayed as <EMPTY> in the Editor. To create one, right-click in a segment, and select "**Set empty translation**". The entry **Remove translation** in the same pop up menu also allows to delete the existing translation of the current segment. You achieve the same by clearing the target segment and pressing Enter.
Insert the best fuzzy match	OmegaT inserts the translation of the string most similar to the current source, if it is above the similarity threshold that you have selected in this dialog. The prefix (per default empty) can be used to tag translations, done via fuzzy matches. If you add a prefix (for instance [fuzzy]), you can trace those translations later to see they are correct.

The check boxes in the lower half of the dialog window serve the following purpose:

Attempt to convert numbers when inserting a fuzzy match

If this option is checked, when a fuzzy match is inserted, either manually or automatically, OmegaT attempts to convert the numbers in the fuzzy matches according to the source contents. There are a number of restrictions:

- The source segment and the fuzzy matches must contain the same list of numbers
- The numbers must be exactly the same between the source and the target matches.
- Only integers and simple floats (using the period as a decimal character, e.g. 5.4, but not 5,4 or 54E-01) are considered.

Allow the translation to be equal to source

Documents for translation may contain trade marks, names or other proper nouns that will be the same in translated documents. There are two strategies for segments that contain only such invariable text.

You can decide not to translate such segments at all. OmegaT will then report these segments as not translated. This is the default. The alternative is to enter a translation that is identical to the source text. OmegaT is able to recognize that you have done this. To make this possible, go to Options → Editing Behavior... and check the box **Allow translation to be equal to source**.

Export the segment to text files

The text export function exports data from within the current OmegaT project to plain text files. The data are exported when the segment is opened. The files appear in the /script subfolder in the OmegaT user files folder, and include:

- The content of the segment source text (source.txt).
- The content of the segment target text (target.txt).
- The text highlighted by the user, when **Ctrl+Shift+C** is pressed or Edit > Export Selection is selected (selection.txt).

The content of the files is overwritten either when a new segment is opened (source.txt and target.txt) or when a new selection is exported (selection.txt). The files are unformatted plain text files. The whole process can be steered and controlled via Tck/Tcl-based scripting. See Using the OmegaT text export function [http://www.omegat.org/en/howtos/text_export.html] for specifics, examples and suggestions.

Go To Next Untranslated Segment stops where there is at least one alternative translation

If we want to avoid any mis-translations in case of segments with several possible target contents, checking this check box will cause **Go To Next Untranslated Segment**stop on the next such segment, irrespective of whether it has already been translated or not.

Chapter 11. Working with plain text

1. Default encoding

Plain text files - in most cases files with a txt extension - contain just textual information and offer no clearly defined way to inform the computer which language they contain. The most that OmegaT can do in such a case, is to assume that the text is written in the same language the computer itself uses. This is no problem for files encoded in Unicode using a 16 bit character encoding set. If the text is encoded in 8 bits, however, one can be faced with the following awkward situation: instead of displaying the text, for Japanese characters...

OmegaT とは、コンピューターを利用した翻訳ツールです。

...the system will display it like this for instance:

OmegaTВ∆ВЌБАГRГУГsГЕБ[Г^ВрЧШЧрВµВљЦ|ЦуГсБ[ГЛВ≈ВЈБВ

The computer, running OmegaT, has Russian as the default language, and thus shows the characters in the Cyrillic alphabet and not in Kanji.

2. The OmegaT solution

There are basically three ways to address this problem in OmegaT. They all involve the application of file filters in the **Options** menu.

Change the encoding of your files to Unicode	open your source file in a text editor that correctly interprets its encoding and save the file in **"UTF-8"** encoding. Change the file extension from .txt to .utf8. OmegaT will automatically interpret the file as a UTF-8 file. This is the most common-sense alternative, sparing you problems in the long run.
Specify the encoding for your plain text files	- i.e. files with a .txt extension - : in the **Text files** section of the file filters dialog, change the **Source File Encoding** from <auto> to the encoding that corresponds to your source .txt file, for instance to .jp for the above example.
Change the extensions of your plain text source files	for instance from .txt to .jp for Japanese plain texts: in the **Text files** section of the file filters dialog, add new **Source Filename Pattern** (*.jp for this example) and select the appropriate parameters for the source and target encoding

OmegaT has by default the following short list available to make it easier for you to deal with some plain text files:

- .txt files are automatically (<auto>) interpreted by OmegaT as being encoded in the computer's default encoding.
- .txt1 files are files in ISO-8859-1, covering most **Western Europe** languages.
- .txt2 files are files in ISO-8859-2, that covers most **Central and Eastern Europe** languages
- .utf8 files are interpreted by OmegaT as being encoded in UTF-8 (an encoding that covers almost all languages in the world).

You can check that yourself by selecting the item **File Filters** in the menu **Options**. For example, when you have a Czech text file (very probably written in the **ISO-8859-2** code) you just need to change the extension .txt to .txt2 and OmegaT will interpret its contents correctly. And of course, if you wish to be on the safe side, consider converting this kind of file to Unicode, i.e. to the .utf8 file format.

Chapter 12. Working with formatted text

Formatting information present in the source file usually needs to be reproduced in the target file. The in-line formatting information made possible by the supported formats (in particular DocBook, HTML, XHTML, Open Document Format(ODF) and Office Open XML (MS Office 2007 and later) at the time of writing) is presented as tags in OmegaT. Normally tags are ignored when considering the similarity between different texts for matching purposes. Tags reproduced in the translated segment will be present in the translated document.

1. Formatting tags

Tag naming:

The tags consist of one to three characters and a number. Unique numbering allows tags, corresponding to each other to be grouped together and differentiates between tags, that have the same shortcut character, but are in fact different. The shortcut characters used try to reflect the underlying meaning of the tag (e.g. b for bold, i for italics, etc.)

Tag numbering:

Tags are numbered incrementally by tag group. "Tag groups" in this context are a single tag (such as <i0> and </i0>). Within a segment, the first group (pair or single) receives the number 0, the second the number 1 etc. The first example below has 3 tag groups (a pair, a single, and then another pair), the second example has one group only (a pair).

Pairs and singles:

Tags are always either singles or paired. Single tags indicate formatting information that does not affect the surrounding text (an extra space or line break for example).

<b0><Ctr+N></b0>, <br1><b2><Enter></b2><segment 2132>

<br1> is a single tag and does not affect any surrounding text. Paired tags usually indicate style information that applies to the text between the opening tag and the closing tag of a pair. <b0> and </b0> below are paired and affect the text log.txt. Note that the opening tag must always come before the corresponding closing tag:

<Log file (<b0>log.txt</b0>) for tracking operations and errors.<segment 3167>

OmegaT creates its tags before the process of sentence segmenting. Depending upon the segmenting rules, the pair of tags may get separated into two consecutive segments and the tag validation will err on the side of caution and mark the two segments.

2. Tag operations

Care must be exercised with tags. If they are accidentally changed, the formatting of the final file may be corrupted. The basic rule is that the sequence of tags must be preserved in the same order. However, it is possible, if certain rules are strictly followed, to deviate from this basic rule.

Tag duplication:

To duplicate tag groups, just copy them in the position of your choice. Keep in mind that in a pair group, the opening tag must come before the closing tag. The formatting represented by the group you have duplicated will be applied to both sections.

Example:

<b0>This formatting</b0> is going to be duplicated here.<segment 0001>

After duplication:

<b0>This formatting</b0> has been <b0>duplicated here</b0>.<segment 0001>

Tag group deletion:

To delete tag groups, just remove them from the segment. Keep in mind that a pair group must have both its opening and its closing tag deleted to ensure that all traces of the formatting are properly erased, otherwise the translated file may become corrupted. By deleting a tag group you will remove the related formatting from the translated file.

Example:

<b0>This formatting</b0> is going to be deleted.<segment 0001>

After deletion:

This formatting has been deleted.<segment 0001>

3. Tag group nesting

Modifying tag group order may result in the nesting of a tag group within another tag group. This is acceptable, provided the enclosing group totally encloses the enclosed group. In other words, when moving paired tags, ensure that both the opening and the closing tag are both either inside or outside other tag pairs, or the translated file may be corrupted and fail to open.

Example:

<b0>Formatting</b0> <b1>one</b1> is going to be nested inside formatting zero.<segment 0001>

After nesting:

<b0>Formatting <b1>one</b1></b0> has been nested inside formatting zero.<segment 0001>

4. Tag group overlapping

Overlapping is the result of bad manipulations of tag pairs and is guaranteed to result in formatting corruption and sometimes in the translated file not opening at all.

Example:

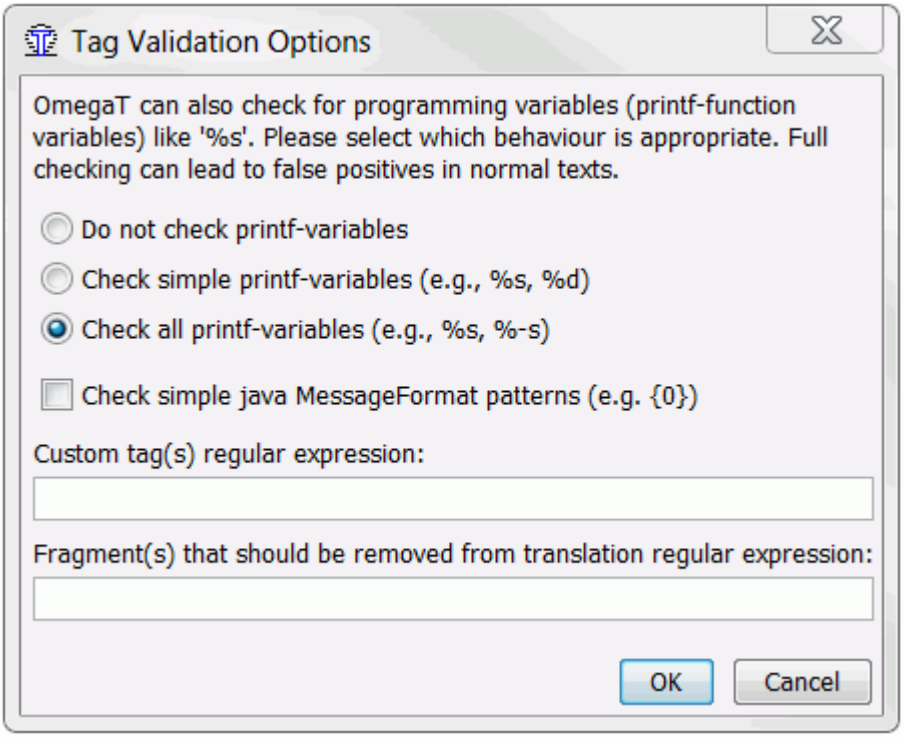
<b0>Formatting</b0> <b1>one</b1> is going to be messed up.<segment 0001>

After a bad manipulation:

<b0>Formatting <b1>one</b0> </b1>is very messed up now.<segment 0001>

5. Tag validation options

To customize the work with tags, one can set down some of the rules in the Options > Tag validation... window:


Tag Validation Options
OmegaT can also check for programming variables (printf-function variables) like '%s'. Please select which behaviour is appropriate. Full checking can lead to false positives in normal texts.
Do not check printf-variables
Check simple printf-variables (e.g., %s, %d)
Check all printf-variables (e.g., %s, %-s)
Check simple java MessageFormat patterns (e.g. {0})
Custom tag(s) regular expression:
Fragment(s) that should be removed from translation regular expression:
OK
Cancel

The behaviour, stated here, applies to all the source files and not just to some of the file types, like formatted text.

- **Printf variables - do not check, check simple, check all**

  OmegaT can check that programming variables (like %s for instance) in the source exist in the translation. You can decide not to check at all, check for simple printf variables (like %s %d etc) or for print variables of all types.

- **Check simple java MessageFormat patterns**

  Activating this check box will cause OmegaT to check if simple java MessageFormat tags (like {0}) are processed correctly.

- **Custom tag(s) regular expression**

  A regular expression entered here will cause OmegaT treat the detected instances as customer tags. It checks that the number of tags and their order is identical, just like it is the case for omegat-tags.

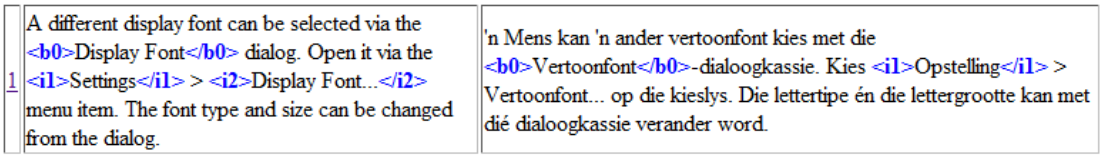
- **Fragment(s) that should be removed from the translation regular expression**

  One can enter a regular expression for unwanted contents in the target. Any matches in the target segment will then be painted red, i.e. easy to identify and correct. When looking for fuzzy matches the remove pattern is ignored. A fixed penalty of 5 is added if the removed part does not match some other segment, so the match does not show up as 100%

6. Tag group validation

The validate tags function detects changes to tag sequences (whether deliberate or accidental), and shows the affected segments. Launching this function – Ctrl+T - opens a window containing all segments in the file containing suspected broken or bad tags in the translation. Repairing the tags and recreating the target documents is easy with the validate tags function. The window that opens when Ctrl+T is pressed features a 3-column table with a link to the segment, the original segment and the target segment

Figure 12.1. Tag validation entry


1
A different display font can be selected via the <b0>Display Font</b0> dialog. Open it via the <i1>Settings</i1> > <i2>Display Font...</i2> menu item. The font type and size can be changed from the dialog.
'n Mens kan 'n ander vertoonfont kies met die <b0>Vertoonfont</b0>-dialoogkassie. Kies <i1>Opstelling</i1> > Vertoonfont... op die kieslys. Die lettertipe én die lettergrootte kan met dié dialoogkassie verander word.

The tags are highlighted in bold blue for easy comparison between the original and the translated contents. Click on the link to activate the segment in the Editor. Correct the error if necessary (in the case above it is the missing <i2></i2> pair) and press Ctrl+T to return to the tag validation window to correct other errors. Tag errors are tag sequences in the translation in which the same tag order and number as in the original segment is not reproduced. Some tag manipulations are necessary and are benign, others will cause problems when the translated document is created.

7. Hints for tags management

Simplify the original text

Tags generally represent formatting in some form of the original text. Simplifying the original formatting greatly contributes to reducing the number of tags. Where circumstances permit, unifying used fonts, font sizes, colors, etc. should be considered, as it could simplify the translation and reduce the potential for tag errors. Read the tag operations section to see what can be done with tags. Remember that if you find tags a problem in OmegaT and formatting is not extremely relevant for the current translation, removing tags may be the easiest way out of problems.

Pay extra attention to tag pairs

If you need to see tags in OmegaT but do not need to retain most of the formatting in the translated document you are free not to include tags in the translation. In this case pay extra attention to tag pairs since deleting one side of the pair but forgetting to delete the other is guaranteed to corrupt your document's formatting. Since tags are included in the text itself, it is possible to use segmentation rules to create segments with fewer tags. This is an advanced feature and some experience is required in order for it to be applied properly.

OmegaT is not yet able to detect mistakes in formatting fully automatically, so it will not prompt you if you make an error or change formatting to fit your target language better. Sometimes, however, your translated file may look strange, and – in the worst case – may even refuse to open.

Chapter 13. Translation memories

1. Translation memories in OmegaT

1.1. tmx folders - location and purpose

OmegaT projects can have translation memory files - i.e. files with the extension tmx - in five different places:

omegat folder

The omegat folder contains the project_save.tmx and possibly a number of backup TMX files. The project_save.tmx file contains all the segments that have been recorded in memory since you started the project. This file always exists in the project. Its contents will always be sorted alphabetically by the source segment.

main project folder

The main project folder contains 3 tmx files, project_name-omegat.tmx, project_name-level1.tmx and project_name-level2.tmx (project_name being the name of your project).

- The level1 file contains only textual information.
- The level2 file encapsulates OmegaT specific tags in correct tmx tags so that the file can be used with its formatting information in a translation tool that supports tmx level 2 memories, or OmegaT itself.
- The OmegaT file includes OmegaT specific formatting tags so that the file can be used in other OmegaT projects

These files are copies of the file project_save.tmx, i.e. of the project's main translation memory, excluding the so-called orphan segments. They carry appropriately changed names, so that its contents still remain identifiable, when used elsewhere, for instance in the tm subfolder of some other project (see below).

tm folder

The /tm/ folder can contain any number of ancillary translation memories - i.e. tmx files. Such files can be created in any of the three varieties indicated above. Note that other CAT tools can export (and import as well) tmx files, usually in all three forms. The best thing of course is to use OmegaT-specific TMX files (see above), so that the in-line formatting within the segment is retained.

The contents of translation memories in the tm subfolder serve to generate suggestions for the text(s) to be translated. Any text, already translated and stored in those files, will appear among the fuzzy matches, if it is sufficiently similar to the text currently being translated.

If the source segment in one of the ancillary TMs is identical to the text being translated, OmegaT acts as defined in the Options → Editing Behavior... dialog window. For instance (if the default is accepted), the translation from the ancillary TM is accepted and prefixed with *[fuzzy]*, so that the translator can review the translations at a later stage and check whether the segments tagged this way, have been translated correctly (see the Editing behavior chapter) .

It may happen, that translation memories, available in the tm subfolder, contain segments with identical source text, but differing targets. TMX files are read sorted by their names and segments within a given TMX file line by line. The last segment with the identical source text will thus prevail (Note: of course it makes more sense to avoid this to happen in the first place).

Note that the TMX files in the tm folder can be compressed with gzip.

tm/auto folder

If it is clear from the very start, that translations in a given TM (or TMs) are all correct, one can put them into the **tm/auto** folder and avoid confirming a lot of *[fuzzy]* cases. This will effectively **pre-translate** the source text: all the segments in the source text, for which translations can be found in those "auto" TMs, will land in the main TM of the project without any user intervention.

tm/penalty-xxx folders

Sometimes, it is useful to distinguish between high-quality translation memories and those that are, because of the subject matter, client, revision status, etc., less reliable. For translation memories in folders with a name "penalty-xxx" (with xxx between 0 and 100), matches will be degraded according to the name of the folder: a 100% match in any of TMs, residing in a folder called Penalty-30 for instance, will be lowered to a 70% match. The penalty applies to all three match percentages: matches 75, 80, 90 will in this case be lowered to 45, 50, 60.

Optionally, you can let OmegaT have an additional tmx file (OmegaT-style) anywhere you specify, containing all translatable segments of the project. See pseudo-translated memory below.

Note that all the translation memories are loaded into memory when the project is opened. Back-ups of the project translation memory are produced regularly (see next chapter), and project_save.tmx is also saved/updated when the project is closed or loaded again. This means for instance that you do not need to exit a project you are currently working on if you decide to add another ancillary TM to it: you simply reload the project, and the changes you have made will be included.

The locations of the various different translation memories for a given project are user-defined (see Project dialog window in Project properties)

Depending on the situation, different strategies are thus possible, for instance:

several projects on the same subject: keep the project structure, and change source and target folders (Source = source/order1, target = target/order1 etc). Note that you segments from order1, that are not present in order2 and other subsequent jobs, will be tagged as orphan segments; however, they will still be useful for getting fuzzy matches.

several translators working on the same project: split the source files into source/Alice, source/Bob... and allocate them to team members (Alice, Bob ...). They can then create their own projects and, deliver their own project_save.tmx, when finished or when a given milestone has been reached. The project_save.tmx files are then collected and possible conflicts as regards terminology for instance get resolved. A new version of the master TM is then created, either to be put in team members' *tm/auto*subfolders or to replace their project_save.tmx files. The team can also use the same subfolder structure for the target files. This allows them for instance to check at any moment, whether the target version for the complete project is still OK

1.2. tmx backup

As you translate your files, OmegaT stores your work continually in project_save.tmx in the project's /omegat subfolders.

OmegaT also backups translation memory to project_save.tmx.YEARMMDDHHNN.bak in the same subfolder whenever a project is opened or reloaded. YEAR is 4-digit year, MM is a month, DD day of the month, HH and NN are hours and minutes when the previous translation memory was saved.

If you believe you have lost translation data, follow the following procedure:

1. Close the project
2. Rename the current project_save.tmx file (e.g. to project_save.tmx.temporary)
3. Select the backup translation memory that is most likely - e.g. the most recent one, or the last version from the day before) to contain the data you are looking for
4. Copy it to project_save.tmx
5. Open the project

1.3. tmx files and language

Tmx files contain translation units, made of a number of equivalent segments in several languages. A translation unit comprises at least two translation unit variants (TUV). Either can be used as the source or target.

The settings in your project indicate which is the source and which the target language. OmegaT thus takes the TUV segments corresponding to the project's source and target language codes and uses them as the source and target segments respectively. OmegaT recognizes the language codes using the following two standard conventions :

- 2 letters (e.g. JA for Japanese), or
- 2- or 3-letter language code followed by the 2-letter country code (e.g. EN-US - See Appendix A, *Languages - ISO 639 code list* for a partial list of language and country codes).

If the project language codes and the tmx language codes fully match, the segments are loaded in memory. If languages match but not the country, the segments still get loaded. If neither the language code not the country code match, the segments will be ignored.

TMX files can generally contain translation units with several candidate languages. If for a given source segment there is no entry for the selected target language, all other target segments are loaded, regardless of the language. For instance, if the language pair of the project is DE-FR, it can be still be of some help to see hits in the DE-EN translation, if there's none in the DE-FR pair.

1.4. Orphan segments

The file project_save.tmx contains all the segments that have been translated since you started the project. If you modify the project segmentation or delete files from the source, some matches may appear as **orphan strings** in the Match Viewer: such matches refer to segments that do not exist any more in the source documents, as they correspond to segments translated and recorded before the modifications took place.

2. Reusing translation memories

Initially, that is when the project is created, the main TM of the project, project_save.tmx is empty. This TM gradually becomes filled during the translation. To speed up this process, existing translations can be reused. If a given sentence has already been translated once, and translated correctly, there is no need for it to be retranslated. Translation memories may also contain reference translations: multinational legislation, such as that of the European Community, is a typical example.

When you create the target documents in an OmegaT project, the translation memory of the project is output in the form of three files in the root folder of your OmegaT project (see the above description). You can regard these three tmx files (-omegat.tmx, -level1.tmx and -level2.tmx) as an "export translation memory", i.e. as an export of your current project's content in bilingual form.

Should you wish to reuse a translation memory from a previous project (for example because the new project is similar to the previous project, or uses terminology which might have been used before), you can use these translation memories as "input translation memories", i.e. for import into your new project. In this case, place the translation memories you wish to use in the */tm* or */tm*/auto folder of your new project: in the former case you will get hits from these translation memories in the fuzzy matches viewer, and in the latter case these TMs will be used to pre-translate your source text.

By default, the /tm folder is below the project's root folder (e.g. .../*MyProject/tm*), but you can choose a different folder in the project properties dialog if you wish. This is useful if you frequently use translation memories produced in the past, for example because they are on the same subject or for the same customer. In this case, a useful procedure would be:

- Create a folder (a "repository folder") in a convenient location on your hard drive for the translation memories for a particular customer or subject.
- Whenever you finish a project, copy one of the three "export" translation memory files from the root folder of the project to the repository folder.
- When you begin a new project on the same subject or for the same customer, navigate to the repository folder in the Project > Properties > Edit Project dialog and select it as the translation memory folder.

Note that all the tmx files in the /tm repository are parsed when the project is opened, so putting all different TMs you may have on hand into this folder may unnecessarily slow OmegaT down. You may even consider removing those that are not required any more, once you have used their contents to fill up the project-save.tmx file.

2.1. Importing and exporting translation memories

OmegaT supports imported tmx versions 1.1-1.4b (both level 1 and level 2). This enables the translation memories produced by other tools to be read by OmegaT. However, OmegaT does not fully support imported level 2 tmx files (these store not only the translation, but also the formatting). Level 2 tmx files will still be imported and their textual content can be seen in OmegaT, but the quality of fuzzy matches will be somewhat lower.

OmegaT follows very strict procedures when loading translation memory (tmx) files. If an error is found in such a file, OmegaT will indicate the position within the defective file at which the error is located.

Some tools are known to produce invalid tmx files under certain conditions. If you wish to use such files as reference translations in OmegaT, they must be repaired, or OmegaT will report an error and fail to load them. Fixes are trivial operations and OmegaT assists troubleshooting with the related error message. You can ask the user group for advice if you have problems.

OmegaT exports version 1.4 TMX files (both level 1 and level 2). The level 2 export is not fully compliant with the level 2 standard, but is sufficiently close and will generate correct matches in other translation memory tools supporting TMX Level 2. If you only need textual information (and not formatting information), use the level 1 file that OmegaT has created.

2.2. Creating a translation memory for selected documents

In case translators need like to share their TMX bases while excluding some of their parts or including just translations of certain files, sharing the complete ProjectName-omegat.tmx is

out of question. The following recipee is just one of the possibilities, but simple enough to follow and without any dangers for the assets.

- Create a project, separate for other projects, in the desired language pair, with an appropriate name - note that the TMXs created will include this name.
- Copy the documents, you need the translation memory for, into the source folder of the project.
- Copy the translation memories, containing the translations of the documents above, into tm/auto subfolder of the new project.
- Start the project. Check for possible Tag errors with **Ctrl+T** and untranslated segments with **Ctrl+U**. To check everything is as expected, you may press **Ctrl+D** to create the target documents and check their contents.
- When you exit the project. the TMX files in the main project folder (see above) now contain the translations in the selected language pair, for the files, you have copied into the source folder. Copy them to a safe place for future referrals.
- To avoid reusing the project and thus possibly polluting future cases, delete the project folder or archive it away from your workplace.

2.3. Sharing translation memories

In cases where a team of translators is involved, translators will prefer to share common translation memories rather than distribute their local versions.

OmegaT interfaces to SVN and Git, two common team software versioning and revision control systems (RCS), available under an open source license. In case of OmegaT complete project folders - in other words the translation memories involved as well as source folders, project settings etc - are managed by the selected RCS. see more in Chapter

2.4. Using TMX with alternative language pairs

There may be cases where you have done a project with e.g. Dutch sources, and a translation in say English. Then you need a translation in e.g. Chinese, but your translator does not understand Dutch; she, however, understands perfectly English. In this case NL-EN translation memory can serve as a go-between to help generate NL-ZH translation.

The solution in our example is to copy the existing translation memory into the tm subfolder and rename it ZH_CN.tmx to indicate the target language of the tmx. The translator will be shown English translations for source segments in Dutch and use them to create the Chinese translation.

Important: the supporting TMX must be renamed XX_YY.tmx, where XX_YY is the target language of the tmx, for instance to ZH_CN.tmx in the example above. The project and TMX source languages should of course be identical - NL in our example. Note that only one TMX for a given language pair is possible, so if several translation memories should be involved, you will need to merge them all into the XX_YY.tmx.

3. Sources with existing translations

Some types of source files (for instance PO, TTX, etc.) are bilingual, i.e. they serve both as a source and as a translation memory. In such cases, an existing translation, found in the file, is included in the project_save.tmx. It is treated as a default translation, if no match has been found, or as an alternative translation, in case the same source segment exists, but with a target text. The result will thus depend on the order in which the source segments have been loaded.

All translations from source documents are also displayed in the Comment pane, in addition to the Match pane. In case of PO files, a 20% penalty applied to the alternative translation (i.e., a 100% match becomes an 80% match). The word [Fuzzy] is displayed on the source segment.

When you load a segmented TTX file, segments with source = target will be included, if "Allow translation to be equal to source" in Options → Editing Behavior... has been checked. This may be confusing, so you may consider unchecking this option in this case.

4. Pseudo-translated memory

Note

Of interest for advanced users only!

Before segments get translated, you may wish to pre-process them or address them in some other way than is possible with OmegaT. For example, if you wish to create a pseudo-translation for testing purposes, OmegaT enables you to create an additional tmx file that contains all segments of the project. The translation in this tmx can be either

- translation equals source (default)
- translation segment is empty

The tmx file can be given any name you specify. A pseudo-translated memory can be generated with the following command line parameters:

java -jar omegat.jar --pseudotranslatetmx=<filename> [pseudotranslatetype=[equal|empty]]

Replace <filename> with the name of the file you wish to create, either absolute or relative to the working folder (the folder you start OmegaT from). The second argument --pseudotranslatetype is optional. Its value is either equal (default value, for source=target) or empty (target segment is empty). You can process the generated tmx with any tool you want. To reuse it in OmegaT rename it to *project_save.tmx* and place it in the omegat-folder of your project.

5. Upgrading translation memories

Very early versions of OmegaT were capable of segmenting source files into paragraphs only and were inconsistent when numbering formatting tags in HTML and Open Document files. OmegaT can detect and upgrade such tmx files on the fly to increase fuzzy matching quality and leverage your existing translation better, saving you the work of doing this manually.

A project's tmx will be upgraded only once, and will be written in upgraded form into the project-save.tmx; legacy tmx files will be upgraded on the fly each time the project is loaded. Note that in some cases changes in file filters in OmegaT may lead to totally different segmentation; as a result, you will have to upgrade your translation manually in such rare cases.

Chapter 14. Source segmentation

Translation memory tools work with textual units called segments. OmegaT has two ways to segment a text: by paragraph or by sentence segmentation (also referred to as “rule-based segmentation”). In order to select the type of segmentation, select Project → Properties... from the main menu and tick or untick the check box provided. Paragraph segmentation is advantageous in certain cases, such as highly creative or stylistic translations in which the translator may wish to change the order of entire sentences; for the majority of projects, however, sentence segmentation is a choice to be preferred, since it delivers better matches from previous translations. If sentence segmentation has been selected, you can setup the rules by selecting Options → Segmentation...from the main menu.

Dependable segmentation rules are already available for many languages, so it is likely that you will not need to get involved with writing your own segmentation rules. On the other hand this functionality can be very useful in special cases, where you can increase your productivity by tuning the segmentation rules to the text to be translated.

Warning: because the text will segment differently after filter options have been changed, so you may have to start translating from scratch. At the same time the previous valid segments in the project translation memory will turn into orphan segments. If you change segmentation options when a project is open, you must reload the project in order for the changes to take effect.

OmegaT uses the following sequence of steps:

Structure level segmentation	OmegaT first parses the text for structure-level segmentation. During this process it is only the structure of the source file that is used to produce segments. For example, text files may be segmented on line breaks, empty lines, or not be segmented at all. Files containing formatting (ODF documents, HTML documents, etc.) are segmented on the block-level (paragraph) tags. Translatable object attributes in XHTML or HTML files can be extracted as separate segments.
Sentence level segmentation	After segmenting the source file into structural units, OmegaT will segment these blocks further into sentences.

1. Segmentation rules

The process of segmenting can be pictured as follows: the cursor moves along the text, one character at a time. At each cursor position rules, consisting of a **Before** and **After** pattern, are applied in their given order to see if any of the **Before** patterns are valid for the text on the left and the corresponding **After** pattern for the text on the right of the cursor. If the rule matches, either the cursor moves on without inserting a segment break (for an exception rule) or a new segment break is created at the current cursor position (for the break rule).

The two types of rules behave as follows:

Break rule
: Separates the source text into segments. For example, "*Did it make sense? I was not sure*." should be split into two segments. For this to happen, there should be a break rule for "?", when followed by spaces and a capitalized word. To define a rule as a break rule, tick the Break/Exception check box.

Exception rule
: specify what parts of text should NOT be separated. In spite of the period, "*Mrs. Dalloway* " should not be split in two segments, so an exception rule should be established for Mrs (and for Mr, for Dr, for prof etc), followed by a period. To define a rule as an exception rule, leave the Break/Exception check box unticked.

The predefined break rules should be sufficient for most European languages and Japanese. In view of the flexibility, you may consider defining more exception rules for your source language in order to provide more meaningful and coherent segments.

2. Rule priority

All segmentation rule sets for a matching language pattern are active and are applied in the given order of priority, so rules for specific language should be higher than default ones. For example, rules for Canadian French (FR-CA) should be set higher than rules for French (FR.*), and higher than Default (.*) ones. Thus, when translating from Canadian French the rules for Canadian French - if any - will be applied first, followed by the rules for French and lastly, by the Default rules.

3. Creating a new rule

Major changes to the segmentation rules should be generally avoided, especially after completion of the first draft, but minor changes, such as the addition of a recognized abbreviation, can be advantageous.

In order to edit or expand an existing set of rules, simply click on it in the top table. The rules for that set will appear in the bottom half of the window.

In order to create an empty set of rules for a new language pattern click **Add** in the upper half of the dialog. An empty line will appear at the bottom of the upper table (you may have to scroll down to see it). Change the name of the rule set and the language pattern to the language concerned and its code (see Appendix A, *Languages - ISO 639 code list* for a list of language codes). The syntax of the language pattern conforms to regular expression syntax. If your set of rules handles a language-country pair, we advise you to move it to the top using the **Move Up** button.

Add the **Before** and **After** patterns. To check their syntax and their applicability, it is advisable to use tools which allow you to see their effect directly. See the chapter on Regular expressions. A good starting point will always be the existing rules.

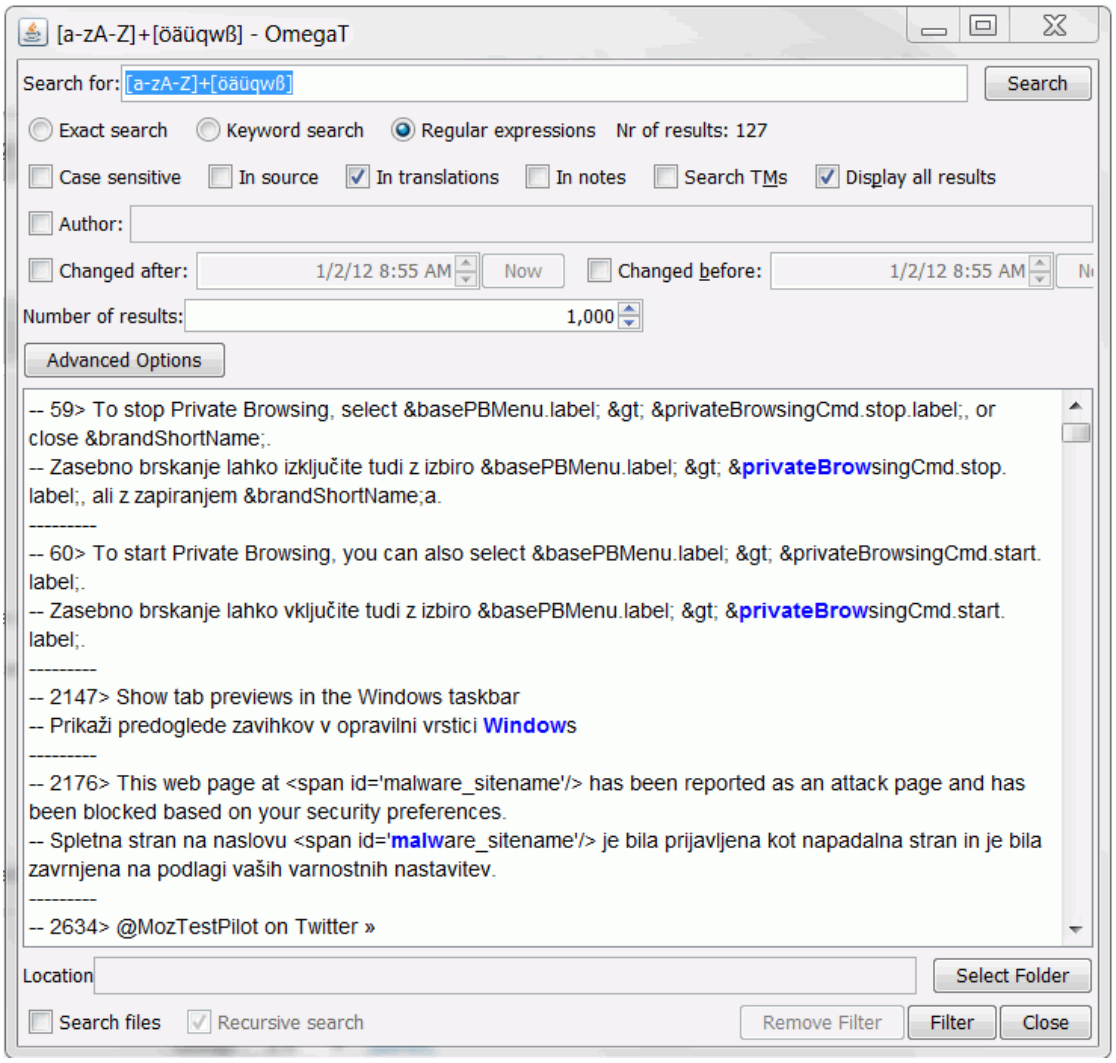
4. A few simple examples

Intention	**Before**	**After**	**Note**
Set the segment start after a period ('.') followed by a space, tab ...	\.	\s	"\." stands for the period character. "\s" means any white space character (space, tab, new page etc.)
Do not segment after Mr.	Mr\.	\s	This an exception rule, so the rule check box must not be ticked
Set a segment after "#" (Japanese period)	#		Note that after is empty
Do not segment after M. Mr. Mrs. and Ms.	Mr??s??\.	\s	Exception rule - see the use of ? in regular expressions

Chapter 15. Searches

1. Search window

Figure 15.1. Text search Window


[a-zA-Z]+[öäüqwß] - OmegaT
Search for: [a-zA-Z]+[öäüqwß]
Search
Exact search
Keyword search
Regular expressions
Nr of results: 127
Case sensitive
In source
In translations
In notes
Search TMs
Display all results
Author:
Changed after:
1/2/12 8:55 AM
Now
Changed before:
1/2/12 8:55 AM
Number of results:
1,000
Advanced Options
-- 59> To stop Private Browsing, select &basePBMenu.label; > &privateBrowsingCmd.stop.label;, or close &brandShortName;.
-- Zasebno brskanje lahko izključite tudi z izbiro &basePBMenu.label; > &privateBrowsingCmd.stop.label;, ali z zapiranjem &brandShortName;a.

-- 60> To start Private Browsing, you can also select &basePBMenu.label; > &privateBrowsingCmd.start.label;.
-- Zasebno brskanje lahko vključite tudi z izbiro &basePBMenu.label; > &privateBrowsingCmd.start.label;.

-- 2147> Show tab previews in the Windows taskbar
-- Prikaži predoglede zavihkov v opravilni vrstici Windows

-- 2176> This web page at has been reported as an attack page and has been blocked based on your security preferences.
-- Spletna stran na naslovu je bila prijavljena kot napadalna stran in je bila zavrnjena na podlagi vaših varnostnih nastavitev.

-- 2634> @MozTestPilot on Twitter »
Location
Select Folder
Search files
Recursive search
Remove Filter
Filter
Close

Note

The above search finds all target segments that contain characters, rather improbable in the target - i.e. Slovenian - language.

1.1. Searching

OmegaT provides a powerful range of search functions. Open the Search window with **Ctrl**+**F** and enter the word or phrase you wish to search for in the Search for box. Alternatively, in the Editor window, select a word or phrase in the editing field (target text of the current segment) and hit **Ctrl**+**F**. The word or phrase is entered in the Search for box automatically. You can have several Search windows open at the same time, but close them when they are no longer needed so that they do not clutter your desktop.

1.2. Search location

Searches are performed by default in both the source and target sections of the project, but you can limit your search to just the source or the target text. You may search in

- the project only (default setting)
- both the project and the reference translation memories in /tm/ (check *Search TMs)*
- a single file or a folder containing a set of files (check *Search files)*

When searching through files (as opposed to translation memories), OmegaT restricts the search to files in source file formats. Consequently, although OmegaT is quite able to handle tmx files, it does not include them in the Search files search. If you turn on **Advanced Options**, you can combine the search with the name of the author of the translation and time of the last change made.

2. Using wild cards

In both exact and keyword searches, the wild card search characters '*' and '?' can be used. They have the meaning, familiar to Word users:

- '*' matches zero or more characters, from the current position in a given word to its end. The search term 'run*' for example would match words 'run', 'runs' and 'running'.
- '?' matches any single character. For instance, 'run?' would match the word 'runs' and 'runn' in the word 'running'.

The matches will be displayed in bold blue. Note that '*' and '?' have special meaning in regular expressions, so wild card search, as described here, applies to exact and keyword search only (see below).

3. Search methods and options

Select the method using the radio buttons. The following search methods are available:

exact search	Search for segments containing the exact string you specified. An exact search looks for a phrase, i.e. if several words are entered, they are found only if they occur in exactly that sequence. Searching for open file will thus find all occurrences of the string *open file*, but not *file opened* or *open input file*.
keyword search	Search for segments containing all keywords you specified, in any order. Select keyword search to search for any number of individual full words, in any order. OmegaT displays a list of all segments containing all of the words specified. Keyword searches are similar to a search "with all of the words" in an Internet search engine such as Google (AND logic). Using keyword search with *open file* will thus find all occurrences of the string *open file,* as well as *file opened, open input file, file may not be safe to open*, etc.
regular expressions	The search string will be treated as a regular expression. The search string - [a-zA-Z]+[öäüqwß] - in the example above for instance looks for words in the target segment, containing questionable characters from German keyboard. Regular expressions are a powerful way to look for instances of a string. See more in the chapter Regular Expressions.

Additionally to one of the methods above you can select the following:

- **case sensitive**: the search will be performed for the exact string specified; i.e. capitalization is observed.
- **in source:** search in the source segments
- **in translations:** search in the target segments

- **in notes:** search in notes to segments
- **search TMs**: translation memories in the tm folder are included in the search.
- **display all results:** clearing this check box will case OmegaT show just the first occurrence of a hit, if several hits exist

If you click on the button Advanced options additional criteria (author of the translation, date translated etc) as shown in the above image can be selected.

4. Search results display

Pressing the search button after entering a string in the search field displays all the segments in the project that include the entered string. As OmegaT handles identical segments as one single entity, only the first unique segment is shown. The segments are displayed in order of appearance in the project. Translated segments are displayed with the original text at the top and the translated text at the bottom, untranslated segments are displayed as the source only.

Clicking on a segment opens it in the Editor for modifications. You can then switch back to the Search window for the next segment found, for instance to check and, if necessary, correct the terminology.

You may have several Search windows open at the same time. You can quickly see their contents by looking at their title: it will contain the search term used.

5. Filter entries in editor according to search

For easier navigation in the search result set, you can apply the search to the editor. Press the **Filter** button on the bottom to limit the shown entries in the editor window to those that match the current search. You can use normal navigation to go to e.g. the next (untranslated) segment that matches the search criteria.

NB:

- A search may be limited to 1000 items, so if you search on a common phrase, the editor then shows only those 1000 matching entries, and not all entries that match the search criteria.
- A file might have no matching entries, so it will show empty.
- If a search removes duplicates, those duplicates will not be in the Editor.

To remove a filter, press the **Remove filter** button, or reload a project.

Chapter 16. Regular expressions

The regular expressions (or regex for short) used in searches and segmentation rules are those supported by Java. Should you need more specific information, consult the Java Regex documentation [http://download.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html]. See additional references and examples below.

Note

This chapter is intended for advanced users, who need to define their own variants of segmentation rules or devise more complex and powerful key search items.

Table 16.1. Regex - Flags

The construct	... matches the following
(?i)	Enables case-insensitive matching (by default, the pattern is case-sensitive).

Table 16.2. Regex - Character

The construct	... matches the following
x	The character x, except the following...
\uhhhh	The character with hexadecimal value 0xhhhh
\t	The tab character ('\u0009')
\n	The newline (line feed) character ('\u000A')
\r	The carriage-return character ('\u000D')
\f	The form-feed character ('\u000C')
\a	The alert (bell) character ('\u0007')
\e	The escape character ('\u001B')
\cx	The control character corresponding to x
\0n	The character with octal value 0n (0 <= n <= 7)
\0nn	The character with octal value 0nn (0 <= n <= 7)
\0mnn	The character with octal value 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	The character with hexadecimal value 0xhh

Table 16.3. Regex - Quotation

The construct	...matches the following
\	Nothing, but quotes the following character. This is required if you would like to enter any of the meta characters !$()*+.<>?[\]^{|} to match as themselves.
\\	For example, this is the backslash character
\Q	Nothing, but quotes all characters until \E
\E	Nothing, but ends quoting started by \Q

Table 16.4. Regex - Classes for Unicode blocks and categories

The construct	...matches the following
\p{InGreek}	A character in the Greek block (simple block [http://download.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html#ubc])
\p{Lu}	An uppercase letter (simple category [http://download.oracle.com/javase/1.5.0/docs/api/java/util/regex/Pattern.html#ubc])
\p{Sc}	A currency symbol
\P{InGreek}	Any character except one in the Greek block (negation)
[\p{L}&&[^\p{Lu}]]	Any letter except an uppercase letter (subtraction)

Table 16.5. Regex - Character classes

The construct	...matches the following
[abc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)

Table 16.6. Regex - Predefined character classes

The construct	...matches the following
.	Any character (except for line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

Table 16.7. Regex - Boundary matchers

The construct	...matches the following
^	The beginning of a line
$	The end of a line
\b	A word boundary
\B	A non-word boundary

Table 16.8. Regex - Greedy quantifiers

The construct	...matches the following
X?	X, once or not at all
X*	X, zero or more times
X+	X, one or more times

Note

greedy quantifiers will match as much as they can. For example, *a+?* will match the aaa in *aaabbb*

Table 16.9. Regex - Reluctant (non-greedy) quantifiers

The construct	...matches the following
X??	X, once or not at all
X*?	X, zero or more times
X+?	X, one or more times

Note

non-greedy quantifiers will match as little as they can. For example, *a+?* will match the first *a* in *aaabbb*

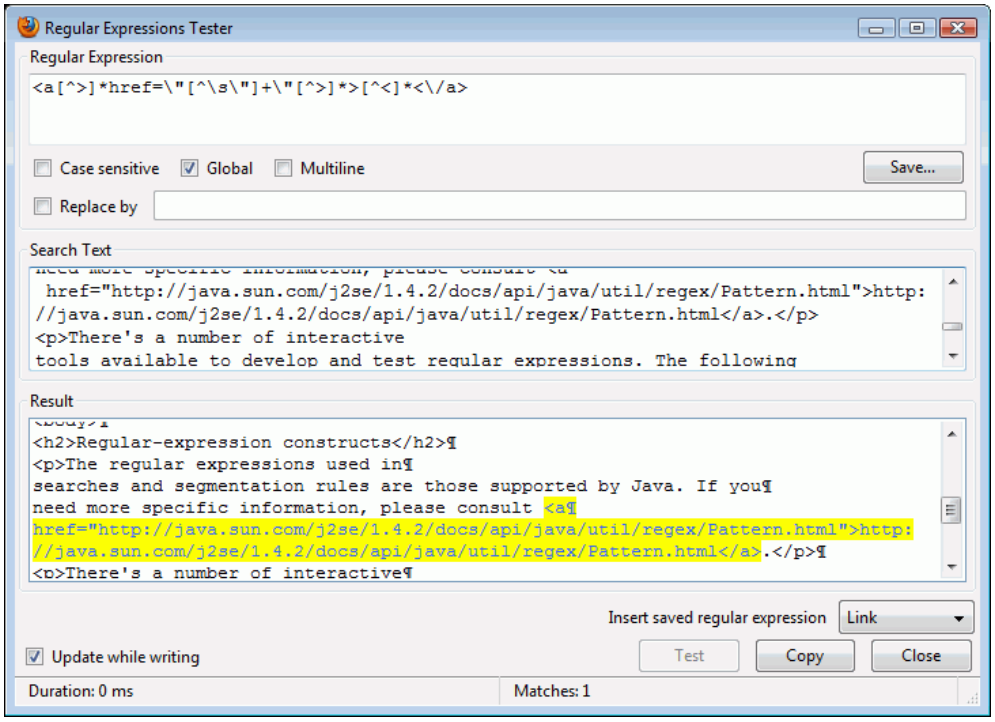
Table 16.10. Regex - Logical operators

The construct	...matches the following
XY	X followed by Y
X|Y	Either X or Y
(XY)	XY as a single group

1. Regex tools and examples of use

A number of interactive tools are available to develop and test regular expressions. They generally follow much the same pattern (for an example from the Regular Expression Tester see below): the regular expression (top entry) analyzes the search text (Text box in the middle) , yielding the hits, shown in the result Text box.

Figure 16.1. Regex Tester



See The Regex Coach [http://weitz.de/regex-coach/] for Windows,Linux, FreeBSD versions of a stand-alone tool. This is much the same as the above example.

A nice collection of useful regex cases can be found in OmegaT itself (see Options > Segmentation). The following list includes expressions you may find useful when searching through the translation memory:

Table 16.11. Regex - Examples of regular expressions in translations

Regular expression	Finds the following:
(\b\w+\b)\s\1\b	double words
[\.,]\s*[\.,]+	comma or a period, followed by spaces and yet another comma or period
\. \s+$	extra spaces after the period at the end of the line
\s+a\s+[aeiou]	English: words, beginning with vowels, should generally be preceded by "an", not "a"
\s+an\s+[^aeiou]	English: the same check as above, but concerning consonants ("a", not "an")
\s{2,}	more than one space
\.[A-Z]	Period, followed by an upper-case letter - possibly a space is missing between the period and the start of a new sentence?
\bis\b	search for “is”, not “this” or “isn't” etc.

Chapter 17. Dictionaries

1. How to download and install dictionaries

Dictionaries in OmegaT are based on the StarDict or on the Lingvo DSL format. To install the necessary files for the StarDict format, proceed as follows:

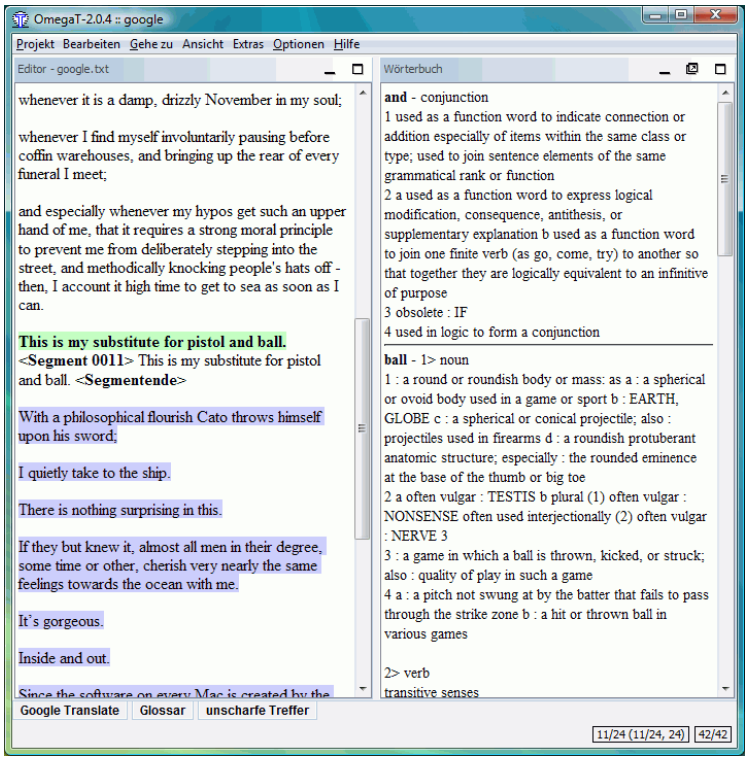
- Search for the required language combination - for instance on the StarDict Platform [http://stardict.sourceforge.net/index.php] above.
- Download the file - it should be a tarball archive (extension *tar.bz* or *tar.bz2*).
- Use untar utility (or its equivalent, for instance winrar in Windows) to extract its contents into the project folder "Dictionary". There should be three files, with extensions *dz, idx* and *ifo*.

Note that in addition to "source-target" dictionaries you can, using the Dictionary feature, obtain access to information such as:

- Webster's Revised Unabridged Dictionary (1913)
- Longman Dictionary of Contemporary English
- The Britannica Concise Encyclopedia
- etc...

Some of the dictionaries have no strings attached - i.e. are "Free to use", and others, like the selection above, are under the GPL license. The following example shows Merriam Webster 10th dictionary "in action":

Figure 17.1. Merriam Webster dictionary - use


OmegaT-2.0.4 :: google
Projekt Bearbeiten Gehe zu Ansicht Extras Optionen Hilfe
Editor - google.txt
Wörterbuch
whenever it is a damp, drizzly November in my soul;
whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet;
and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off - then, I account it high time to get to sea as soon as I can.
This is my substitute for pistol and ball.
<Segment 0011> This is my substitute for pistol and ball. <Segmentende>
With a philosophical flourish Cato throws himself upon his sword;
I quietly take to the ship.
There is nothing surprising in this.
If they but knew it, almost all men in their degree, some time or other, cherish very nearly the same feelings towards the ocean with me.
It's gorgeous.
Inside and out.
and - conjunction
1 used as a function word to indicate connection or addition especially of items within the same class or type; used to join sentence elements of the same grammatical rank or function
2 a used as a function word to express logical modification, consequence, antithesis, or supplementary explanation b used as a function word to join one finite verb (as go, come, try) to another so that together they are logically equivalent to an infinitive of purpose
3 obsolete : IF
4 used in logic to form a conjunction
ball - 1> noun
1 : a round or roundish body or mass: as a : a spherical or ovoid body used in a game or sport b : EARTH, GLOBE c : a spherical or conical projectile; also : projectiles used in firearms d : a roundish protuberant anatomic structure; especially : the rounded eminence at the base of the thumb or big toe
2 a often vulgar : TESTIS b plural (1) often vulgar : NONSENSE often used interjectionally (2) often vulgar : NERVE 3
3 : a game in which a ball is thrown, kicked, or struck; also : quality of play in such a game
4 a : a pitch not swung at by the batter that fails to pass through the strike zone b : a hit or thrown ball in various games
2> verb
transitive senses
Google Translate
Glossar
unscharfe Treffer
11/24 (11/24, 24)
42/42

2. Problems with dictionaries

- Check that your dict files are in the correct folder (or in a subfolder below it). Check in Project → Properties (**Ctrl+E**).
- Does the folder contain three files of the same name, with extensions? If only one file is present, check its extension. If it is *tar.bz*, you have forgotten to unpack (untar) it.

Chapter 18. Glossaries

Glossaries are files created and updated manually for use in OmegaT.

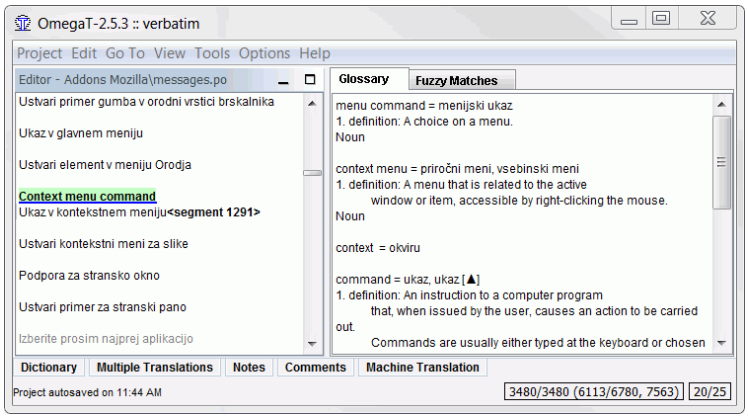
If an OmegaT project contains one or more glossaries, any terms in the glossary which are also found in the current segment will be automatically displayed in the Glossary viewer.

You define its location and name in the project properties dialog. The extension must be .txt(if not, it will be added). The location of the file must be within the /glossaryfolder, but it can be in a deeper folder (e.g., glossary/sub/glossary.txt). The file does not need to exist when setting it, it will be created (if necessary) when adding a glossary entry. If the file already exists, no attempt is done to verify the format or the character set of the file: the new entries will always be in tab-separated format and UTF-8. As the existing content will not be touched, damage to an existing file would be limited.

1. Usage

To use an existing glossary, simply place it in the /glossary folder after creating the project. OmegaT automatically detects glossary files in this folder when a project is opened. Terms in the current segment which OmegaT finds in the glossary file(s) are displayed in the Glossary pane:

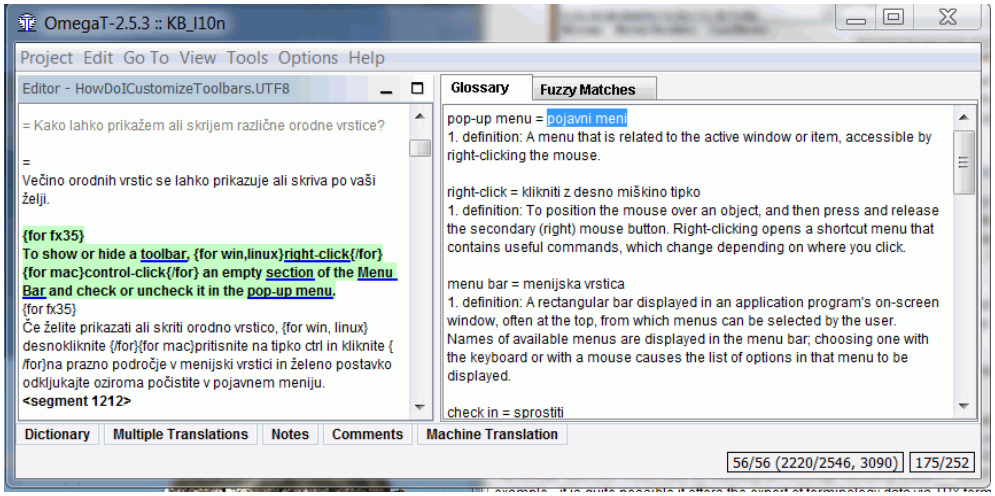
Figure 18.1. Glossary pane


OmegaT-2.5.3 :: verbatim
Project Edit Go To View Tools Options Help
Editor - Addons Mozilla\messages.po
Ustvari primer gumba v orodni vrstici brskalnika
Ukaz v glavnem meniju
Ustvari element v meniju Orodja
Context menu command
Ukaz v kontekstnem meniju<segment 1291>
Ustvari kontekstni meni za slike
Podpora za stransko okno
Ustvari primer za stranski pano
Izberite prosim najprej aplikacijo
Glossary
Fuzzy Matches
menu command = menijski ukaz
1. definition: A choice on a menu.
Noun
context menu = priročni meni, vsebinski meni
1. definition: A menu that is related to the active
window or item, accessible by right-clicking the mouse.
Noun
context = okviru
command = ukaz, ukaz [▲]
1. definition: An instruction to a computer program
that, when issued by the user, causes an action to be carried
out.
Commands are usually either typed at the keyboard or chosen
Dictionary
Multiple Translations
Notes
Comments
Machine Translation
Project autosaved on 11:44 AM
3480/3480 (6113/6780, 7563)
20/25

The word before the = sign is the source term, and its translation is (or are) the words after =. The vocabulary entry can have a comment added. The glossary function only finds exact matches with the glossary entry (e.g. does not find inflected forms etc.). New terms can be added manually to the glossary file(s) during translation, for example in a text editor. Newly added terms will not be recognized once the changes in the text file have been saved.

The source term does not have to be a single-word item, as the next example shows:

Figure 18.2. multiple words entries in glossaries - example


OmegaT-2.5.3 :: KB_I10n
Project Edit Go To View Tools Options Help
Editor - HowDoICustomizeToolbars.UTF8
= Kako lahko prikažem ali skrijem različne orodne vrstice?
=
Večino orodnih vrstic se lahko prikazuje ali skriva po vaši želji.
{for fx35}
To show or hide a toolbar, {for win,linux}right-click{/for} {for mac}control-click{/for} an empty section of the Menu Bar and check or uncheck it in the pop-up menu.
{for fx35}
Če želite prikazati ali skriti orodno vrstico, {for win, linux} desnokliknite {/for}{for mac}pritisnite na tipko ctrl in kliknite { /for}na prazno področje v menijski vrstici in želeno postavko odkljukajte oziroma počistite v pojavnem meniju.
<segment 1212>
Glossary
Fuzzy Matches
pop-up menu = pojavni meni
1. definition: A menu that is related to the active window or item, accessible by right-clicking the mouse.
right-click = klikniti z desno miškino tipko
1. definition: To position the mouse over an object, and then press and release the secondary (right) mouse button. Right-clicking opens a shortcut menu that contains useful commands, which change depending on where you click.
menu bar = menijska vrstica
1. definition: A rectangular bar displayed in an application program's on-screen window, often at the top, from which menus can be selected by the user. Names of available menus are displayed in the menu bar; choosing one with the keyboard or with a mouse causes the list of options in that menu to be displayed.
check in = sprostiti
Dictionary
Multiple Translations
Notes
Comments
Machine Translation
56/56 (2220/2546, 3090)
175/252

The underlined item "pop-up menu" can be found in the glossary pane as "pojavni menu". Highlighting it in the Glossary pane and then rightclicking insets at the cursor position in the target segment.[1]

2. File format

Glossary files are simple plain text files containing three-column, tab-delimited lists with the source and target terms in the first and second columns respectively. The third column can be used for additional information. You can have entries with the target column missing, i.e. just containing the source term and the comment.

Glossary files can be either in system default encoding (and indicated by the extension .tab) or in UTF-8 (the extension .utf8). The Unicode encoding (UTF8) is preferred for obvious reasons. Also supported is the CSV format. This format is the same as the tab separated one: source term, target term. Comment fields are separated by a comma ','. Strings can be enclosed by quotes ", which allows having a comma inside a string:

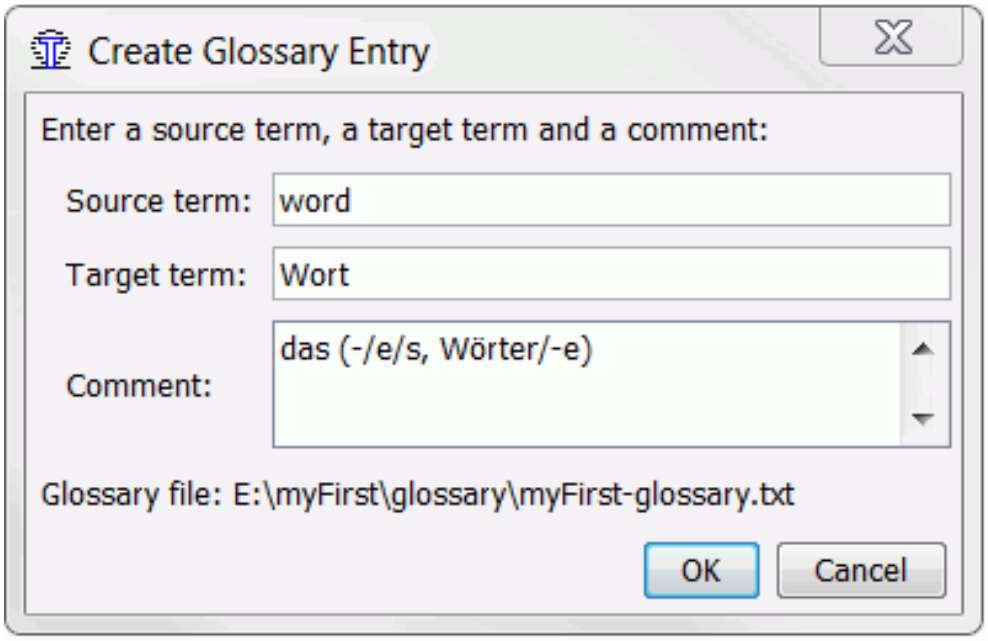
"This is a source term, which contains a comma","c'est un terme, qui contient une virgule"

In addition to the plain text format, TBX format is also supported. TBX - Term Base eXchange - is the open, XML-based standard for exchanging structured terminological data, TBX has been approved as an international standard by LISA and ISO. If you have an existing terminology handling system - MultiTerm for example - it is quite possible it offers the export of terminology data via TBX format. Microsoft Terminology Collection [http://www.microsoft.com/Language/en-US/Terminology.aspx] can be downloaded in nearly 100 languages and can serve as a cornerstone IT glossary.

3. How to create glossaries

The project setting allows one can enter a name for a writable glossary file (see beginning of this chapter). Right-click in the glossary pane or press **Ctrl+Shift+G** to add a new entry. A dialog opens, allowing you to enter the source term, target term and any comments you may have:

[1] Note that in the above case, this is just half (or even less) of the story, as the target language (Slovenian) uses declension. So the inserted "pojavni meni" in the nominative form - has to be changed to "pojavnem meniju" , i.e. to the locative. So it is probably faster to type the term correctly right away without bothering with the glossary and its shortcuts.



The contents of glossary files are kept in memory and are loaded when the project is opened or reloaded. Updating a glossary file is thus rather simple: press **Ctrl+Shift+G** and enter the new term, its translation and any comments you may have (ensuring you press tab between the fields) and save the file. The contents of the glossary pane will be updated accordingly.

The location of the writable glossary file can be set in the Project > Properties ... dialog. The recognized extensions are TXT and UTF8

Note: Of course there are other ways and means to create a simple file with tab delimited entries. Nothing speaks against using Notepad++ on Windows, GEdit on Linux for instance or some spreadsheet program for this purpose: any application, that can handle UTF8 and that can show white space (so that you do not miss the required **TAB** characters) can be used.

4. Using Trados MultiTerm

Data exported from Trados MultiTerm can be used as OmegaT glossaries without further modification, provided they are given the file extension .tab and the source and target term fields are the first two fields respectively. If you export using the system option "Tab-delimited export", you will need to delete the first 5 columns (Seq. Nr, Date created etc). The newer versions of MultiTerm support exporting to TBX format.

5. Common glossary problems

Problem: No glossary terms are displayed - possible causes:

- No glossary file found in the "glossary" folder.
- The glossary file is empty.
- The items are not separated with a TAB character.
- The glossary file does not have the correct extension (.tab or .utf8).
- There is no EXACT match between the glossary entry and the source text in your document - for instance plurals.
- The glossary file does not have the correct encoding.
- There are no terms in the current segment which match any terms in the glossary.

- One or more of the above problems may have been fixed, but the project has not been reloaded.

Problem: In the glossary pane, some characters are not displayed properly

- ...but the same characters are displayed properly in the Editing pane: the extension and the file encoding do not match.

Chapter 19. Machine Translation

1. Introduction

As opposed to user-generated translation memories (as in the case of OmegaT) Machine translation (MT) tools use rule-based linguistic tools to create a translation of the source segment without the need for a translation memory. Statistical learning techniques, based on source and target texts, are used to build a translation model. Machine translation services have been achieving good and steadfastly improving results in research evaluations.

To activate any of the Machine Translation services, go to Options > Machine Translate ... and activate the service desired. Note that they are all web-based: you will have to be on-line if you want to use them.

2. Google Translate

Google Translate is a payable service offered by Google, for translating sentences, web sites and complete texts between an ever-growing number of languages. At the time of writing the list includes more than 50 languages, from Albanian to Yiddish, including of course all the major languages. The current version of the service is based on usage, with the price of 20 USD per million characters at the time of writing.

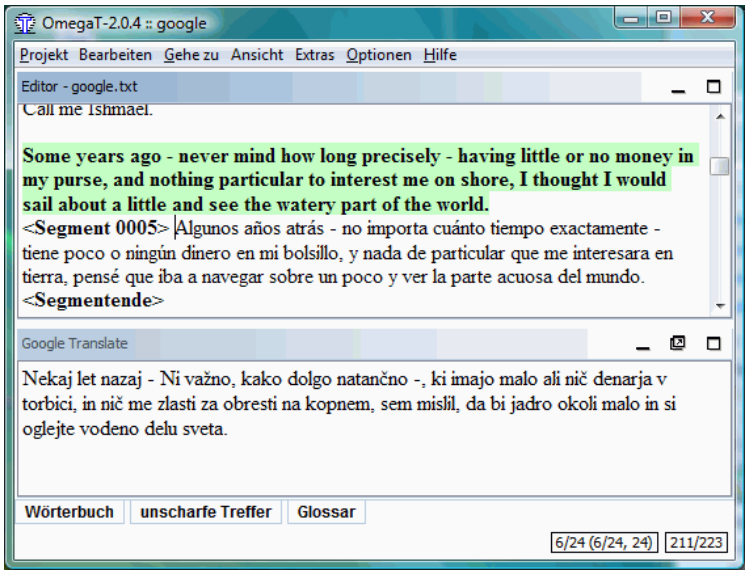
Important: Google Translate API v2 requires billing information for all accounts before you can start using the service (see Pricing and Terms of Service [https://developers.google.com/translate/v2/pricing?hl=en-US] for more). To identify yourself as a valid user for the Google services, you use your private unique key sent to you by Google, when you have registered for the service. See chapter Installing and Running, section Launch command arguments, for details on how to add this key to the OmegaT environment.

The quality of the Google Translate translation depends on one side on the reservoir of target-language texts and the availability of their bilingual versions, on the other hand on the quality of the models built. It is pretty much certain that while the quality may be insufficient in some cases, it will definitely get better with time and not worse.

3. OmegaT users and Google Translate

The OmegaT user is not forced to use Google Translate. If used, neither the user's decision to accept the translation nor the final translation are made available to Google. The following window shows an example of a) the English source b) Spanish and c) Slovenian Google Translate translation.

Figure 19.1. Google Translate - example


OmegaT-2.0.4 :: google
Projekt Bearbeiten Gehe zu Ansicht Extras Optionen Hilfe
Editor - google.txt
Call me Ishmael.
Some years ago - never mind how long precisely - having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world.
<Segment 0005> Algunos años atrás - no importa cuánto tiempo exactamente - tiene poco o ningún dinero en mi bolsillo, y nada de particular que me interesara en tierra, pensé que iba a navegar sobre un poco y ver la parte acuosa del mundo.
<Segmentende>
Google Translate
Nekaj let nazaj - Ni važno, kako dolgo natančno -, ki imajo malo ali nič denarja v torbici, in nič me zlasti za obresti na kopnem, sem mislil, da bi jadro okoli malo in si oglejte vodeno delu sveta.
Wörterbuch
unscharfe Treffer
Glossar
6/24 (6/24, 24)
211/223

The Spanish translation is better than the Slovenian. Note *interesar* and *navegar* in Spanish, are correctly translated as the verbs interest and sail respectively. In the Slovenian version both words have been translated as nouns. It is actually quite probable that the Spanish translation is based at least partially on the actual translation of the book.

Once you have activated the service, a suggestion for the translation will appear in the Machine Translate pane every time a new source segment is opened. If you find the suggestion acceptable, press **Ctrl**+**M** to replace the target part of the opened segment with the suggestion. In the above segment, for instance, **Ctrl**+**M** would replace the Spanish version with the Slovenian suggestion.

If you do not wish OmegaT to send your source segments to Google to get translated, untick the Google Translate menu entry in Options.

Note that nothing but your source segment is sent to the MT service. The online version of Google Translate allows the user to correct the suggestion and send the corrected segment in. This feature, however, is not implemented in OmegaT.

4. Belazar

Belazar [http://belazar.info/] is a Machine language translation tool for the Russian-Belarusian language pair.

5. Apertium

Apertium [http://www.apertium.org/] is a free/open-source machine translation platform, initially aimed at related-language pairs, like CA, ES, GA, PT, OC and FR but recently expanded to deal with more divergent language pairs (such as English-Catalan). Check the web site for the latest list of implemented language pairs.

The platform provides

- a language-independent machine translation engine
- tools to manage the linguistic data necessary to build a machine translation system for a given language pair and
- linguistic data for a growing number of language pairs

Apertium uses a shallow-transfer machine translation engine which processes the input text in stages, as in an assembly line: de-formatting, morphological analysis, part-of-speech

disambiguation, shallow structural transfer, lexical transfer, morphological generation, and re-formatting.

It is possible to use Apertium to build machine translation systems for a variety of language pairs; to that end, Apertium uses simple XML-based standard formats to encode the linguistic data needed (either by hand or by converting existing data), which are compiled using the provided tools into the high-speed formats used by the engine.

6. Machine translation - trouble shooting

If there's nothing appearing in the Machine Translate pane, then check the following:

- Are you online? You need to be online to be able to use an MT tool.
- What is the language pair you need? Check if the selected service offers it.
- Google Translate does not work: have you applied Translate API service [https://developers.google.com/translate/v2/faq]? Note that Google Translate service is not free of charge, see chapter Installing and Running (runtime parameters) for more on that.
- "Google Translate returned HTTP response code: 403 ...": check that the 38-characters key, entered in the pinfo.list file, is correct. Check that Translate API service [https://developers.google.com/translate/v2/faq]has been activated.
- Google Translate does not work: - with the Google API key entered as requested. Check in Options > Machine Translate, that Google Translate V2 is checked.
- Google Translate V2 reports "Bad request" - check the source and target languages for your project. Having no languages defined elicits this kind or a response.

Chapter 20. Spell checker

OmegaT has a built-in spell checker based on the spelling checker used in Apache OpenOffice, LibreOffice, Firefox and Thunderbird. It is consequently able to use the huge range of free spelling dictionaries available for these applications.

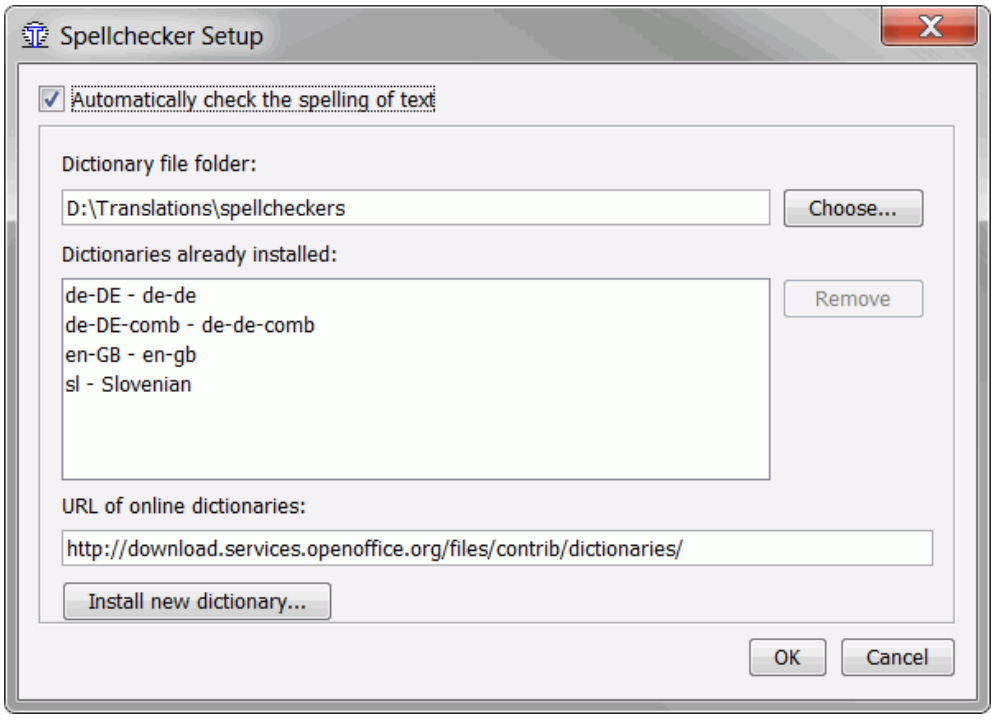
1. Installing spelling dictionaries

Before the spell check function can be used, a suitable dictionary or dictionaries (i.e. for your target language) must be installed. To install spelling dictionaries, follow this procedure:

- In your file manager, create a new folder in a suitable location in which to store spelling dictionaries (D:\Translations\spellcheckers in the example below).
- In OmegaT, select Options > Spell Checking, then click Choose beside the Dictionary file folder field. Navigate to and select the folder you created for dictionaries.
- Place the dictionary files you wish to use in this folder. There are essentially two ways in which you can do this. You can either copy files manually, i.e. from elsewhere on your system, using your file manager; or you can use OmegaT's **"Install new dictionary"** function to provide a list of available dictionaries to select from. Note that the "Install" function requires an Internet connection. The selected languages will then be installed and will eventually appear in your spell checker setup window (this may take a while).

Copying the files manually makes sense if you already have suitable dictionary files on your system, for instance as part of your Apache OpenOffice, LibreOffice, Firefox or Thunderbird installation. It is simpler, however, to look for dictionaries online, using the **URL of online dictionaries** field:

Figure 20.1. Spellchecker setup



Clicking on Install new dictionary button will open the Dictionary installer window, where you can select the dictionaries you want to install.

The names of the files must correspond to the language code of your target language as defined in the project properties dialog (Project > Properties). For example, if you have

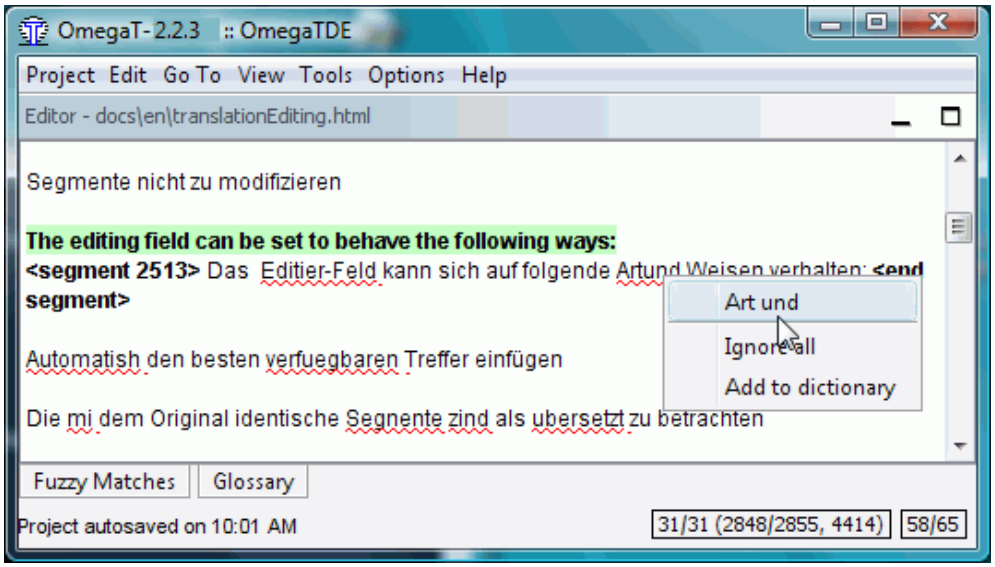
selected ES-MX (Mexican Spanish) as the target language, the dictionary files must be named es_MX.dic and es_MX.aff. If you only have a standard Spanish dictionary available, with file names es_es.dic and es_es.aff for instance, you can copy these files to es_MX.dic and es_MX.aff, and the spelling dictionary will work. Note that this will of course check for the standard (Castillian) rather than for Mexican Spanish.

2. Using spelling dictionaries

There is no need to instruct OmegaT to use a particular spelling dictionary; OmegaT will use the correct language dictionary based upon the language codes of your project. Check however that the language codes are exactly the same: an FR-FR dictionary will not work with an FR target setting, for example. If necessary, edit the file names of the dictionary or change your project's language settings.

To enable the spell checker, select Options > Spell Checking and tick the Automatically check the spelling of text check box (see above).

Figure 20.2. Using spellchecker


OmegaT-2.2.3 :: OmegaTDE
Project Edit Go To View Tools Options Help
Editor - docs\en\translationEditing.html
Segmente nicht zu modifizieren
The editing field can be set to behave the following ways:
<segment 2513> Das Editier-Feld kann sich auf folgende Artund Weisen verhalten: <end segment>
Art und
Ignore all
Add to dictionary
Automatish den besten verfuegbaren Treffer einfügen
Die mi dem Original identische Segnente zind als ubersetzt zu betrachten
Fuzzy Matches
Glossary
Project autosaved on 10:01 AM
31/31 (2848/2855, 4414)
58/65

Right-clicking on an underlined word (Artund in the figure above) opens a drop-down menu listing suggestions for the correction (Art und). You can also instruct the spell checker to ignore all the occurrences of the mis-spelled word, or add it to the dictionary.

3. Hints

If the spell checker is not working, then make sure first that the check box "Automatically check the spelling of text" in the spell checker dialog (Options > Spell checking...) is checked.

Also check that the target language code of your project against the available vocabularies in the setup window. The spell checker uses the target language code to determine the language to be used : if the target language is Brazilian Portuguese (PT_BR), the subfolder with vocabularies must contain the two vocabulary files, called *pt_br.aff* and *pt_br.dic*.

If you have already translated a large body of text, and then realize the target language code of the project does not match the spell checker's language code (you specified pt_BR as the language, but there are no pt_BR vocabularies, for instance) you can simply copy the two corresponding files and rename them (e.g. from *pt_PT.aff* and *pt_PT.dic* to *pt_BR.aff* and *pt_BR.dic*). Of course it is much wiser, to take a short break and download the correct versions of the spell checker.

Note that Remove physically removes the selected vocabularies. If they are used by some other application on your system, they will disappear from that application, too. If, for

whatever reason, you need to do this from time to time, it may make sense to copy the files involved to a different folder, reserved just for use by OmegaT.

Chapter 21. Miscellaneous subjects

1. OmegaT Console Mode

Note

Of interest for advanced users only!

The purpose of the console (i.e. command line) mode is to permit the use of OmegaT as translation tool in a scripting environment. When launched in console mode, no GUI is loaded (it will work therefore on any console) and the given project is automatically translated. An example would be a software project, with GUI localized in a number of languages. Using the console mode, one can make generating a localized interface a part of the build process.

1.1. Prerequisites

To run OmegaT, a valid OmegaT project must be available. The location is irrelevant, since it must be specified explicitly on the command-line at launch.

If you need non-standard settings, the corresponding configuration files (filters.conf and segmentation.conf) must be present. This can be achieved in two ways:

- Run OmegaT normally (with the GUI) and set the settings. If you start OmegaT in console mode, it will use the settings you configured.
- If you are unable to run OmegaT normally (no graphical environment available): copy the settings files from some other OmegaT installation on another machine to a specific folder. The location does not matter, since you can add it to the command line at launch (see below). The relevant files filters.conf and segmentation.conf can be found in the user home folder (E.g. C:\Documents and Settings\%User%\OmegaT under Windows, %user %/.omegat/ under Linux)

1.2. Launching OmegaT in console mode

To launch OmegaT in console mode, additional parameters must be specified at launch. The most important of these is <project-dir>, and optionally --config-dir=<config-dir>. Example:

$> java -jar OmegaT.jar /path/to/project \

--config-dir=/path/to/config-files/ \

--mode=console-translate \

--source-pattern={regexp} \

--tag-validation=[block|warn]

Explanation:

- <project-dir> tells OmegaT where to find the project to be translated. If given, OmegaT launches in console mode and translates the given project.
- --config-dir=<config-dir> enables OmegaT to be instructed in which folder the configuration files are stored. If not specified, OmegaT reverts to default values (the OmegaT folder in the user home folder, or if not available: the current working folder).
- --mode=console-translate OmegaT launches in console mode and translates the given project

- --source-pattern={regexp} The files to be translated can be specified this way. Here is an example of the regular expression: test\.html
- --tag-validation=[abort|warn] On abort, the program is aborted when tag validation finds errors. On warn the errors are printed but the program continues. In all other cases no tag validation is done.

1.3. Quiet option

An extra command line parameter specific to console mode: --quiet. In the quiet mode, less info is logged to the screen. The messages you would usually find in the status bar are not displayed.

Usage: java -jar OmegaT.jar /path/to/project --mode=console-translate --quiet

1.4. Tag validation option

Another extra command line parameter specific to console mode: --tag-validation=[abort|warn]. When this parameter is added, tag validation is done prior to translation/aligning. If the value is abort, then on tag errors the errors are printed and the program stops. If the value is warn then the errors are printed but OmegaT continues.

Usage: java -jar OmegaT.jar /path/to/project --mode=console-translate --tag-validation=abort

2. Automatic Java Properties Aligner

OmegaT can align Java .properties in console mode. If you have the source and the target Properties files for one and the same contents in your language pair, this procedure will create a corresponding tmx file for these contents. Usage:

java -jar OmegaT.jar --mode=console-align /my-project-dir --alignDir=/translatedFiles/

alignDir must contain a translation in the target language of the project. E.g., if the project is EN->FR, alignDir must contain a bundle ending with _fr. The resulting tmx is stored in the omegat folder under the name align.tmx.

3. Font settings

In this dialog one can define the font used by OmegaT in the following windows:

- OmegaT main window (Editor, Match viewer, Glossary viewer)
- Search window
- Tag validation window

The dialog can be accessed via the Options → Font... item in the Main menu. The dialog contains:

- **Font:** drop-down to select one of the fonts available on your machine
- **Size:** edit to change font size
- **Sample text:** field for immediate preview of the selected font

Note: In some cases it may take quite some time for OmegaT to update the display after the font setting has been changed. This is especially the case when a large file containing many segments is open in the editor, and/or slow hardware is used. Note also that some fonts behave better for some language pairs than for others. In particular, if you are

translating between two languages with different alphabets/writing systems (such as Russian and Japanese), select a font that can be used for both.

4. Preventing data loss

OmegaT is a robust application. However, you should take precautions against data loss when using OmegaT, just as with any other application. When you translate your files, OmegaT stores all your progress in the translation memory project_save.tmx that resides in the project's /omegat subfolder.

OmegaT also backs up the translation memory to project_save.tmx.YEARMMDDHHNN.bak in the same subfolder each time a project is opened or reloaded. YEAR is the 4-digit year, MM is the month, DD the day of the month, and HH and NN are the hours and minutes when the previous translation memory was saved.

If you believe that you have lost translation data, you can use the following procedure to restore the project to its most recently saved state, usually not older than approximately 10 minutes or so:

1. close the project
2. rename the current project_save.tmx file (e.g. to project_save.tmx.temporary)
3. select the backup translation memory that is the most likely to contain the data you are looking for
4. rename it project_save.tmx
5. open the project

To avoid losing important data:

- Make regular copies of the file /omegat/project_save.tmx to backup media, such as CD or DVD.
- Until you are familiar with OmegaT, create translated files at regular intervals and check that the translated file contains the latest version of your translation.
- Take particular care when making changes to the files in /source while in the middle of a project. If the source file is modified after you have begun translating, OmegaT may be unable to find a segment that you have already translated.
- Use these Help texts to get started. Should you run into problems, post a message in the OmegaT user group [http://tech.groups.yahoo.com/group/Omegat/]. Do not hesitate to post in the language you feel the most familiar with.

Appendix A. Languages - ISO 639 code list

Please check the ISO 639 Code Tables [http://www.sil.org/ISO639-3/codes.asp] for further and up-to-date information about language codes.

Table A.1. ISO 639-1/639-2 Language code list

Language name	ISO 639-1	ISO 639-2
Abkhaz	ab	abk
Afar	aa	aar
Afrikaans	af	afr
Akan	ak	aka
Albanian	sq	sqi
Amharic	am	amh
Arabic	ar	ara
Aragonese	an	arg
Armenian	hy	hye
Assamese	as	asm
Avaric	av	ava
Avestan	ae	ave
Aymara	ay	aym
Azerbaijani	az	aze
Bambara	bm	bam
Bashkir	ba	bak
Basque	eu	eus
Belarusian	be	bel
Bengali	bn	ben
Bihari	bh	bih
Bislama	bi	bis
Bosnian	bs	bos
Breton	br	bre
Bulgarian	bg	bul
Burmese	my	mya
Catalan	ca	cat
Chamorro	ch	cha
Chechen	ce	che
Chichewa, Chewa, Nyanja	ny	nya
Chinese	zh	zho
Chuvash	cv	chv
Cornish	kw	cor
Corsican	co	cos
Cree	cr	cre
Croatian	hr	hrv

Language name	ISO 639-1	ISO 639-2
Czech	cs	ces
Danish	da	dan
Divehi, Dhivehi, Maldivian	dv	div
Dutch	nl	nld
Dzongkha	dz	dzo
English	en	eng
Esperanto	eo	epo
Estonian	et	est
Ewe	ee	ewe
Faroese	fo	fao
Fijian	fj	fij
Finnish	fi	fin
French	fr	fra
Fula, Fulah, Pulaar, Pular	ff	ful
Galician	gl	glg
Georgian	ka	kat
German	de	deu
Greek, Modern	el	ell
Guaraní	gn	grn
Gujarati	gu	guj
Haitian, Haitian Creole	ht	hat
Hausa	ha	hau
Hebrew (modern)	he	heb
Herero	hz	her
Hindi	hi	hin
Hiri Motu	ho	hmo
Hungarian	hu	hun
Interlingua	ia	ina
Indonesian	id	ind
Interlingue	ie	ile
Irish	ga	gle
Igbo	ig	ibo
Inupiaq	ik	ipk
Ido	io	ido
Icelandic	is	isl
Italian	it	ita
Inuktitut	iu	iku
Japanese	ja	jpn
Javanese	jv	jav
Kalaallisut, Greenlandic	kl	kal
Kannada	kn	kan
Kanuri	kr	kau

Language name	ISO 639-1	ISO 639-2
Kashmiri	ks	kas
Kazakh	kk	kaz
Khmer	km	khm
Kikuyu, Gikuyu	ki	kik
Kinyarwanda	rw	kin
Kirghiz, Kyrgyz	ky	kir
Komi	kv	kom
Kongo	kg	kon
Korean	ko	kor
Kurdish	ku	kur
Kwanyama, Kuanyama	kj	kua
Latin	la	lat
Luxembourgish, Letzeburgesch	lb	ltz
Luganda	lg	lug
Limburgish, Limburgan, Limburger	li	lim
Lingala	ln	lin
Lao	lo	lao
Lithuanian	lt	lit
Luba-Katanga	lu	lub
Latvian	lv	lav
Manx	gv	glv
Macedonian	mk	mkd
Malagasy	mg	mlg
Malay	ms	msa
Malayalam	ml	mal
Maltese	mt	mlt
Māori	mi	mri
Marathi (Marāṭhī)	mr	mar
Marshallese	mh	mah
Mongolian	mn	mon
Nauru	na	nau
Navajo, Navaho	nv	nav
Norwegian Bokmål	nb	nob
North Ndebele	nd	nde
Nepali	ne	nep
Ndonga	ng	ndo
Norwegian Nynorsk	nn	nno
Norwegian	no	nor
Nuosu	ii	iii
South Ndebele	nr	nbl
Occitan	oc	oci

Language name	ISO 639-1	ISO 639-2
Ojibwe, Ojibwa	oj	oji
Old Church Slavonic, Church Slavic, Church Slavonic, Old Bulgarian, Old Slavonic	cu	chu
Oromo	om	orm
Oriya	or	ori
Ossetian, Ossetic	os	oss
Panjabi, Punjabi	pa	pan
Pāli	pi	pli
Persian	fa	fas
Polish	pl	pol
Pashto, Pushto	ps	pus
Portuguese	pt	por
Quechua	qu	que
Romansh	rm	roh
Kirundi	rn	run
Romanian, Moldavian, Moldovan	ro	ron
Russian	ru	rus
Sanskrit (Saṁskṛta)	sa	san
Sardinian	sc	srd
Sindhi	sd	snd
Northern Sami	se	sme
Samoan	sm	smo
Sango	sg	sag
Serbian	sr	srp
Scottish Gaelic, Gaelic	gd	gla
Shona	sn	sna
Sinhala, Sinhalese	si	sin
Slovak	sk	slk
Slovene	sl	slv
Somali	so	som
Southern Sotho	st	sot
Spanish, Castilian	es	spa
Sundanese	su	sun
Swahili	sw	swa
Swati	ss	ssw
Swedish	sv	swe
Tamil	ta	tam
Telugu	te	tel
Tajik	tg	tgk
Thai	th	tha
Tigrinya	ti	tir

Language name	ISO 639-1	ISO 639-2
Tibetan Standard, Tibetan, Central	bo	bod
Turkmen	tk	tuk
Tagalog	tl	tgl
Tswana	tn	tsn
Tonga (Tonga Islands)	to	ton
Turkish	tr	tur
Tsonga	ts	tso
Tatar	tt	tat
Twi	tw	twi
Tahitian	ty	tah
Uighur, Uyghur	ug	uig
Ukrainian	uk	ukr
Urdu	ur	urd
Uzbek	uz	uzb
Venda	ve	ven
Vietnamese	vi	vie
Volapük	vo	vol
Walloon	wa	wln
Welsh	cy	cym
Wolof	wo	wol
Western Frisian	fy	fry
Xhosa	xh	xho
Yiddish	yi	yid
Yoruba	yo	yor
Zhuang, Chuang	za	zha
Zulu	zu	zul

Appendix B. Keyboard shortcuts in the editor

This short text describes key behavior in the editor pane. The term "Move to inside segment" means, that the cursor moves to the beginning of the segment if it was previously before the segment, and to the end of the segment if it was previously after it.

Table B.1. Key behavior in the editor

Key combination	Action
Left:	one char left, but not further than the beginning of segment
Right:	one char right, but not further than the end of segment
Ctrl+Left:	one word left, but not further than the beginning of segment
Ctrl+Right:	one word right, but not further than the end of segment
PgUp:	page up through the document
PgDn:	page down through the document
Home*	move to the beginning of the line in the segment
End*	move to the end of the line in the segment
Ctrl+Home	move to the start of the segment
Ctrl+End	move to the end of the segment
Ctrl+PgUp	move to the start of the document (Mac: **Cmd+PgUp**)
Ctrl+PgDn	move to the end of the document (Mac: **Cmd+PgDn**)
Backspace*	remove char before cursor
Delete*	remove char after cursor
Ctrl+Backspace	remove chars up to the start of the current word (Mac: **Alt+Backspace**)
Ctrl+Delete	remove chars up to the start of next word (Mac: **Alt+Delete**)
Ctrl+Enter	open previous segment (Mac: **Cmd+Enter**)
Ctrl+A	select complete segment (Mac: **Cmd+A**)
Ctrl+Shift+O	RTL-LTR switch

* These keys behave differently when the cursor is outside the editable segment:

- **Home:** cursor to the beginning of the active segment
- **End:** cursor to the end of the active segment
- **Backspace:** nothing
- **Delete:** nothing
- **Any char key,** if clicked outside editable segment, will be ignored.

The "Shift" key doesn't exhibit any special behavior per se: when the "Shift" key is pressed, all keys move the cursor in their usual manner, except in the case of the Shift+Enter combination, that inserts a line break into the text.

System-wide commands Select All (**Ctrl+A**), Paste (**Ctrl+V**), Cut (**Ctrl+X**), copy (**Ctrl+C)**, Insert match **(Ctrl+I**) and Insert source (**Ctrl+Shift+I**) act in principle on the text within the currently open segment only.

Note that you can reassign the shortcuts to your own preferences. See Appendix ShortCut Customization

Appendix C. OmegaT Team Projects

1. Version control - introduction

The collaborative translation offered by OmegaT is based on the functionality of version or revision control, widely used by the software community to maintain control of changes to the code of a program and allow unimpeded collaboration within the development team. OmegaT supports two of the popular version control systems (VCS for short), Apache Subversion [http://subversion.apache.org] (often abbreviated SVN, after the command name svn) and Git [http://git-scm.com/]. The advantages of a VC system for a team of translators can be summarized as follows:

- Several team members can work on the translation project simultaneously without interfering with each other
- They can share common material, like project translation memory and its glossary
- Every three minutes by default, an updated version of data shared is available to the rest of the team
- The system maintains versioning for data shared
- Conflicts - for instance alternative translations of the same segment or glossary entry - can be monitored, resolved and merged

The following terms, to be used in the text below, deserve a short explanation:

- **VCS server -** i.e. SVN or Git server is the environment where the common material is kept and maintained on the net. The server can exist in the local network but in the majority of cases it will be available on internet, i.e. via URL address. One member of the team, the project administrator, needs to be acquainted with handling the server side, i.e. the job of setting up the environment, importing the OmegaT project, assigning the access rights for the team members, resolving the conflicts, etc.
- **VCS client:** To interface with the server an SVN or Git client must be installed on computers of "project managers" involved in the OmegaT project. Very popular clients for Windows environment are TortoiseSVN [http://tortoisesvn.net/] and TortoiseGit [http://code.google.com/p/tortoisegit/]. Other operating systems (Linux, OS X) offer similar packages.
- **repository:** the place where the shared material is saved and maintained, either on a local access network or in Internet. Project members connect with it via their VCS client.
- **checkout:** the operation that creates a working copy from the repository to your local computer. The server keeps the information on checkouts, so that later commits (see below) can be performed in an orderly fashion.
- **commit:** once a new local version of the checked-out material is ready, it can be committed to the repository and thus made available to the rest of the team. The server makes sure that any conflicting changes, due to two members working on the same checked-out contents, will be resolved.
- **administrator:** the person responsible for the creation and maintaining of the repository, i.e. taking care of the server side of the task. To avoid any problems, one person only should have these rights at least initially.
- **user**: a member of the team, collaborating on the common project.

2. Sharing a project using SVN

There are two possibilities to run an SVN server: you can install SVN on your own server or you can use a hosted service. When using an external service you must be aware of the possible

implications in terms of confidentiality, since you are loading the original document on a server outside of your direct control. Alternatively, to avoid this issue you can set a private SVN server, for example if you already have an Apache server that includes the software in question (e.g. VisualSVN).

Once the SVN server is available, project managers must locally install a SVN client, in order to manage the project contents on their computers. For Windows we recommend TortoiseSVN [http://tortoisesvn.net/]. For Mac you can download the client for instance from SourceForge [https://sourceforge.net/projects/macsvn/], For Linux see Subversion Commands and Scripts [www.yolinux.com/TUTORIALS/Subversion.html].

2.1. Creating a repository

The procedure presented here relies on the free SVN server (limited to 2 users) offered by ProjectLocker [http://projectlocker.com/]. Note that the creator of the repository has implicitly the administrator rights for the repository created. Sign in to the site first or - if it is your first time on the site, register for it and note your user name and password for the future projects.

1. Create a new project on ProjectLocker

2. Type the name and description of the repository. (OmegaT and OmegaT SL Localization in the example used here)

3. Choose SVN.

4. Click Create Project

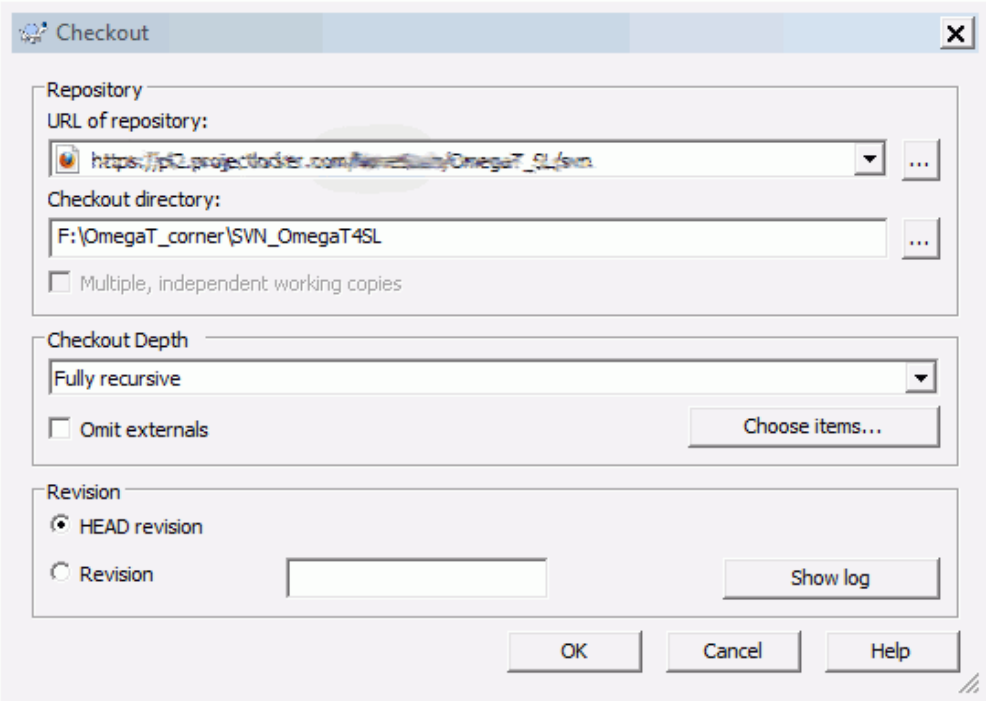
Open the **Projects** view for your account. The URL shown under Project Services will be used by SVN to connect clients to the SVN server you have just established. This is also the place to add members of the team to the project and assign them their rights. Note that the team members have to be registered first, before you can add them to the project (Note: in the free version of ProjectLocker you are allowed only two users per project).

Projects can be managed according to your development style and needs. Similar as in the case of OmegaT projects, you will need to have separate repositories for different language pairs. Within a given language pair it is best to keep different subjects and/or clients as separate repositories as well. The alternative is to have one single repository with subfolders Project1, Project2, etc., and share the common material via common tm, glossary and dictionary folders.

For the example shown here we decided for the one OmegaT project - one single repository for the simplicity reasons.

2.2. Importing the project to SVN repository - Windows

The repository is empty at this moment. You create first an empty client folder on your disk. Create an empty folder, where you will keep your project and right-click on it. Select TortoiseSVN > Checkout. The following dialog appears:



Enter the URL, provided by ProjectLocker, into the field **URL of repository**. Make sure the field **Checkout directory** is correct, i.e. specifies the empty folder you have created, and press **OK**. Once the operation has finished, you can check the said folder: it should now contain a subfolder .svn and a green OK badge on its icon will show, that the contents of the folder are up-to-date:

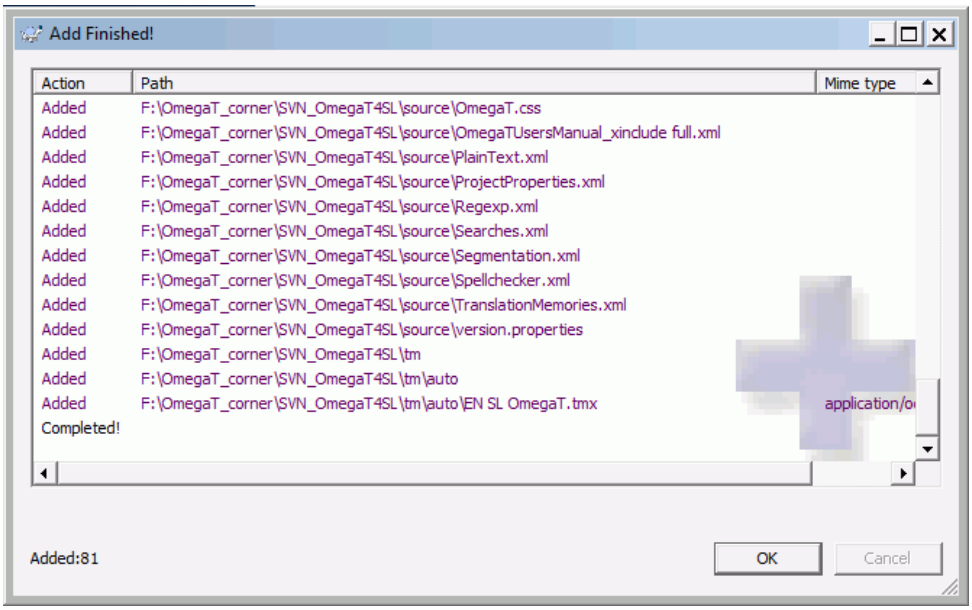
In the next step, we will add the OmegaT files to the local folder. The following files are to be shared among the members of the team and thus have to be included in any case:

- the omegat project file - omegat.project
- the translation memory - omegat\project_save.tmx
- the contents of the source folder
- the project-specific filters definition - omegat\filters.xml

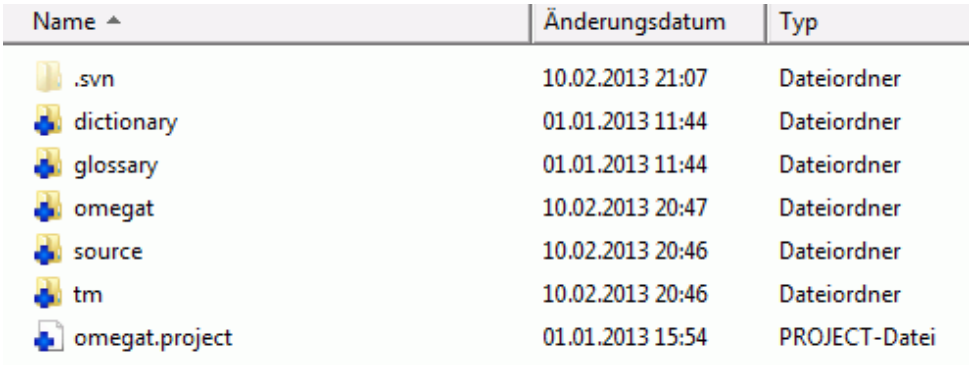
The administrator may decide to include following folders and their contents as well: tm, glossary and dictionary. Also ignored_words.txt and learned_words.txt in the omegat folder may be worth sharing and maintaining on the team level. Avoid in any case adding bak files, project_stats.txt and project_stats_match.txt, in the omegat subfolder, as they would without any need or profit just bloat the repository. You might want to apply the same to the target folder and its contents.

After copying the required files into the checkout folder you will notice that its icon has changed: the green OK badge has changed to a red exclamation sign, signifying the change in the local copy of the repository. The following two steps will bring the server version up to date:

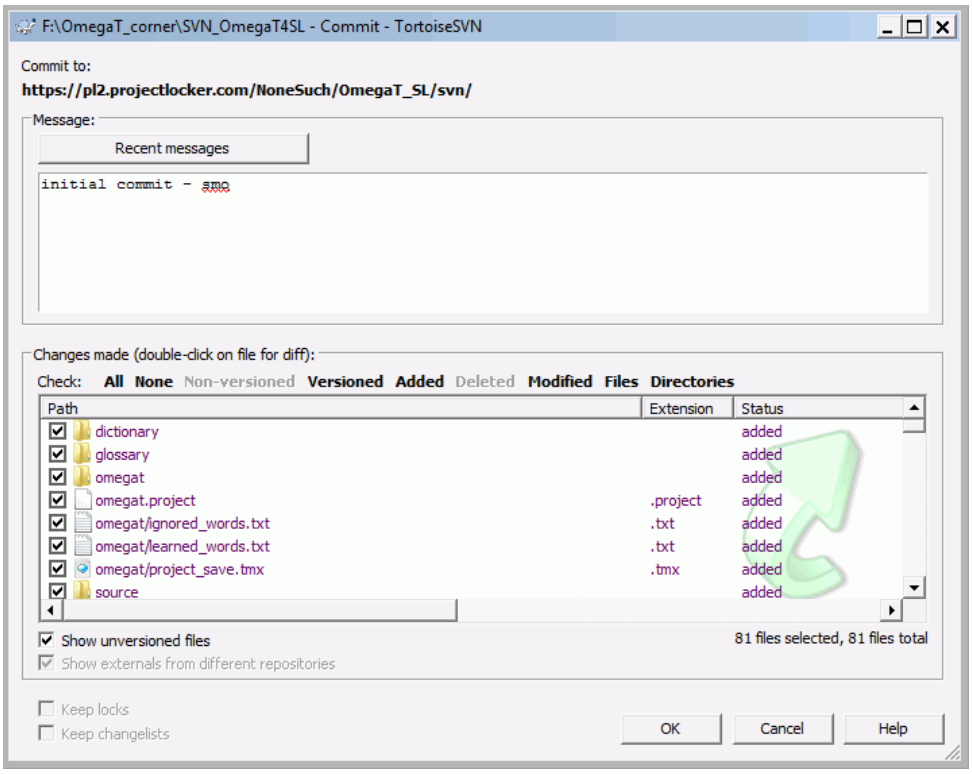
- **add the copied files to the local version of the repository:** right-click on the local checkout folder and select TortoiseSVN > Add. In the dialog that opens, leave all options as per default and click OK. The Add Finished! window, similar to the one below will appear:



The contents of the checkout folder will be marked accordingly:

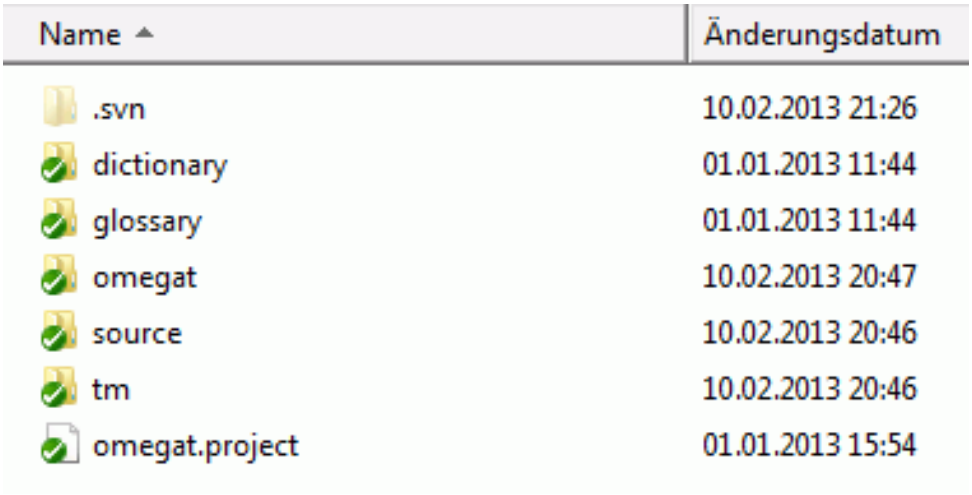


- **commit local changes to the server:** right-click on the local checkout folder and select SVN Commit.... The Commit window - see below opens. Check the changes to be made - i.e. the folders and files added in this case.



Enter an appropriate message into the message window and press OK. The Commit window will open and show the progress of the commit command. It will first commit the current contents to the server repository and then update the local copy of the repository - i.e. the contents of .svn subfolder - so that it is up to date with the latest repository version.

- **update local files from the local repository copy** - the changes received from the server repository reside within the .svn subfolder but not yet in the files and folders themselves. To update the local files, right-click on the checkout folder and select SVN Update. Check the contents of the folder to confirm that the local copy of the repository and the corresponding files and folders correspond to the latest server version:



3. Using the team project in OmegaT

Once the team project is setup, team members only need OmegaT to access the team project. First, they need to use Project > Download Team Project. This will actually do a checkout of the project in a local folder. Credentials are stored, so it isn't needed to enter them each time. Under Linux, if OmegaT is still asking for your credentials, you can checking the Force saving password as plain text checkbox.

For subsequent use, all is needed is opening the project like any other OmegaT project. OmegaT will recognize it is a team project, and will synchronize everything automatically, every three minutes by default.

Appendix D. Tokenizer plugin

1. Introduction

Tokenizers (or stemmers) improve the quality of matches by recognizing inflected words in source and translation memory data. They also improve glossary matching.

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish". This is especially useful in case of languages that use pre- and postfix forms for the stem words. Borrowing an example from Slovenian, here "good" in all possible grammatically correct forms:

- lep, lepa, lepo - singular, masculine, feminine, neutral
- lepši, lepša, lepše . - comparative, nominative, masculine, feminine, neutral, resp. Plural form of the adjective
- najlepših - superlative, plural, genitive for M,F,N

2. Installation and command line use

A tokenizer package adapted from the Lucene project is distributed as an OmegaT plug-in at http://sourceforge.net/projects/omegat-plugins/files/. Download the most recent files (OmegaT-tokenizers_0.4_2-2.1.zip at the time of this writing).

To install the tokenizer, create a folder with the name "plugins" in the folder where OmegaT.jar is found and unpack the above files within that folder.

To run OmegaT with the tokenizer you need to specify which tokenizer you'll use for the source language and which tokenizer you'll use for the target language. The syntax is as follows:

java -jar OmegaT.jar --ITokenizer=[source language tokenizer name] --ITokenizerTarget=[target language tokenizer name]

The tokenizer names are given in the Readme.txt file distributed with the tokenizer files. For example, if you wish to use the Lucene CJK tokenizer in source and the Lucene French tokenizer in target, your command will look like this:

java -jar OmegaT.jar --ITokenizer=org.omegat.plugins.tokenizer.LuceneCJKTokenizer --ITokenizerTarget=org.omegat.plugins.tokenizer.LuceneFrenchTokenizer

3. Mac OS X specifics

If you wish to use the tokenizers with the Mac OS X OmegaT.app package, the tokenizer installation described above applies (right-click on OmegaT.app to find the location of OmegaT.jar), but you'll need to specify the tokenizer names in the info.plist that contains the Java launch options. Follow the instructions above to access the info.plist file and edit it so that it looks as follows for the example we just gave:

```
<key>VMOptions</key>
<string>-Xmx1024M</string>

<key>Arguments</key>
<array>
  <string>--ITokenizer=org.omegat.plugins.tokenizer.LuceneCJKTokenizer</string>
```

```
  <string>--ITokenizerTarget=org.omegat.plugins.tokenizer.LuceneFrenchTokenizer</string>
</array>
```

4. Troubleshooting

To make sure that the tokenizers are being used, open a project and check the log information from the console. With the example above it should look like that:

```
84528: Info: Source tokenizer: org.omegat.plugins.tokenizer.LuceneCJKTokenizer
84528: Info: Target tokenizer: org.omegat.plugins.tokenizer.LuceneFrenchTokenizer
```

The numbers on the left are likely to be different on your system so make sure that the source and target tokenizer names,specified in the start-up options, correspond to what the log is showing. If the tokenizers are not properly launched, the log will look like this:

```
12719: Info: Source tokenizer: org.omegat.core.matching.Tokenizer
12719: Info: Target tokenizer: org.omegat.core.matching.Tokenizer
```

With the Mac OS X OmegaT.app package, double-click on the JavaApplicationStub located in /OmegaT.app/Contents/MacOS/ (see above to access it) to launch OmegaT from the console and get immediate access to the log.

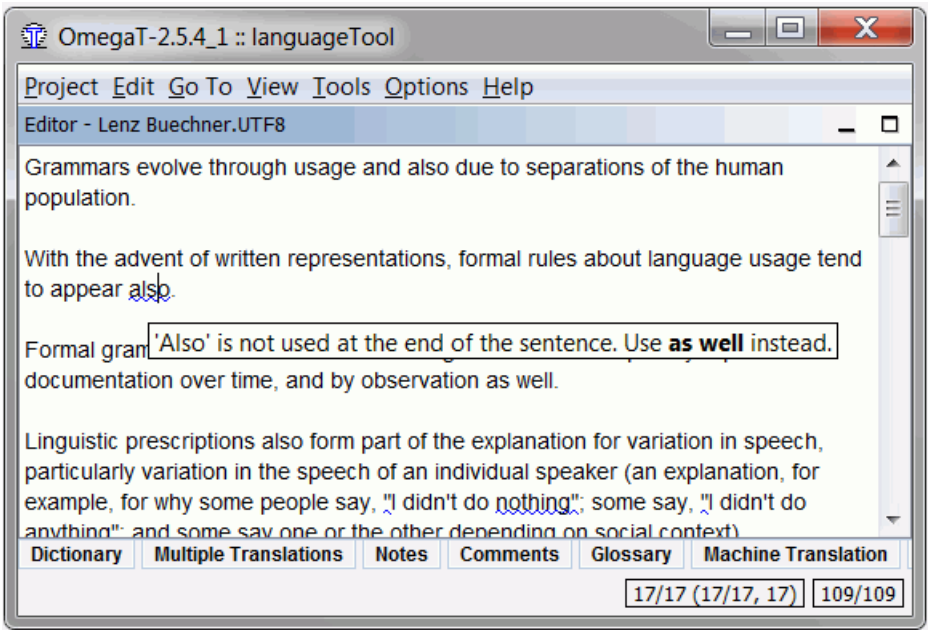
Appendix E. LanguageTool plugin

1. Introduction

LanguageTool [http://www.languagetool.org] is an Open Source style and grammar proofreading software for English, French, German, Polish, Dutch, Romanian, and a number of other languages - see the list of supported languages [http://www.languagetool.org/languages/].

You can think of LanguageTool as a software to detect errors that a simple spell checker cannot detect, e.g. mixing up there/their, no/now etc. It can also detect some grammar mistakes. It does not include spell checking. LanguageTool will find errors for which a rule has been defined in its language-specific configuration files.

Figure E.1. The LanguageTool in OmegaT


OmegaT-2.5.4_1 :: languageTool
Project Edit Go To View Tools Options Help
Editor - Lenz Buechner.UTF8
Grammars evolve through usage and also due to separations of the human population.
With the advent of written representations, formal rules about language usage tend to appear alsb.
Formal gran
'Also' is not used at the end of the sentence. Use as well instead.
documentation over time, and by observation as well.
Linguistic prescriptions also form part of the explanation for variation in speech, particularly variation in the speech of an individual speaker (an explanation, for example, for why some people say, "I didn't do nothing"; some say, "I didn't do anything"; and some say one or the other depending on social context)
Dictionary
Multiple Translations
Notes
Comments
Glossary
Machine Translation
17/17 (17/17, 17)
109/109

2. Installation and Use

The LanguageTool plugin for OmegaT can be downloaded from the OmegaT plugins download site [http://sourceforge.net/projects/omegat-plugins/files]. Unzip the files, and put them all in the "plugins" folder located in OmegaT installation folder, that is the folder that contains the OmegaT.jar. Create the "plugins" folder first, if it does not exist.

Once installed, the plugin will be used automatically by OmegaT, if Options > Language Checker is checked. The rules applied (if any) will depend on the source and target language of the project. When a rule is triggered, the corresponding phrase will be underlined in blue in the Editor (see *also* and *I didn't do nothing* in the picture above). When hovering the mouse over the underlined phrase, an explanation will appear.

Appendix F. Scripting plugin

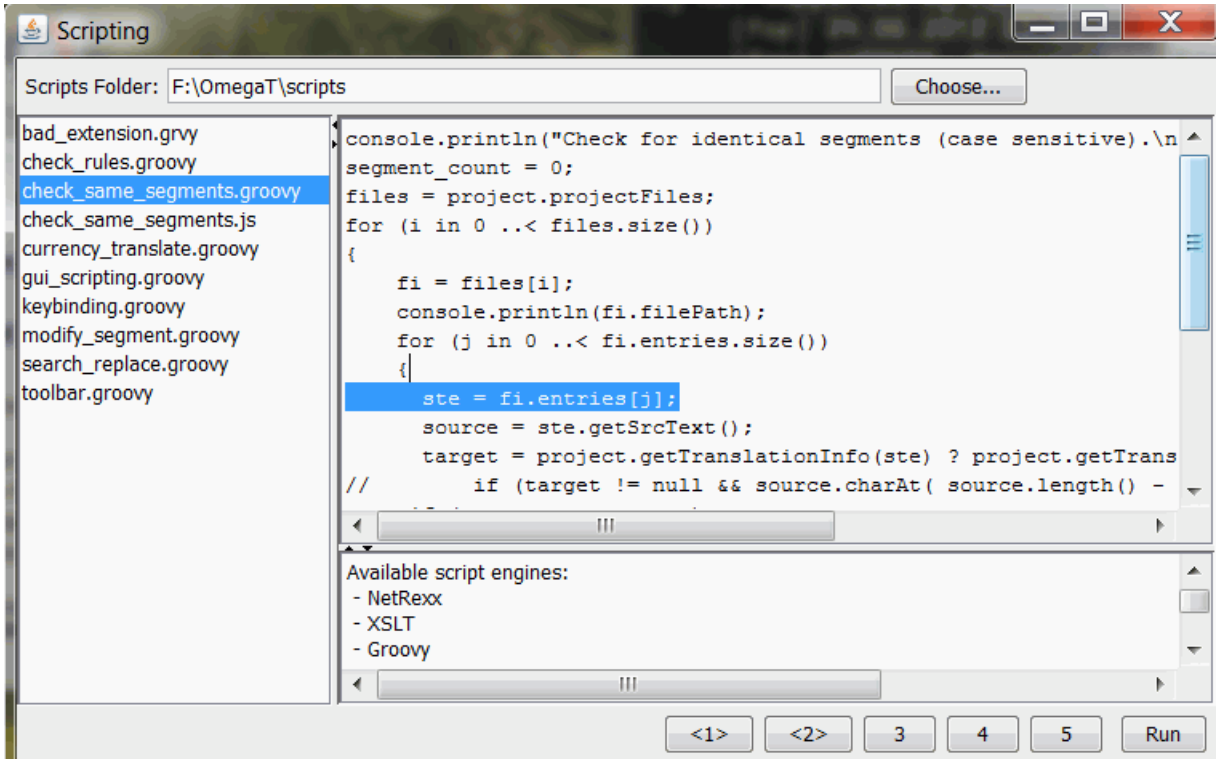
1. Introduction

OmegaT-Scripting allows to run scripts written in different scripting languages in OmegaT.

2. Installation and Use

The Scripting plugin for OmegaT can be downloaded from the OmegaT plugins download site [http://sourceforge.net/projects/omegat-plugins/files]. Unzip the files, and put them all in the "plugins" folder located in OmegaT installation folder, that is the folder that contains the OmegaT.jar. Create the "plugins" folder first, if it does not exist.

This adds a new submenu Tools > Scripting. Below the Scripting submenu , there's five place holders available for scripts. Clicking Scripting opens the Scripting dialog:



The Scripting dialog allows you to load an existing script into the text area and run it against the current opened project. To customize the script feature, do the following:

- Load a script into the editor by clicking on its name in the list on the left panel.
- Right-click on a button from "<1>" to "<5>" in the bottom panel and select "Add". In the above example, two scripts (position 1 and 2) have already been added.
- When you left-click on the number, the selected script will run. You can start the selected macros from the main menu as well by using their entries in the Tools menu or by pressing **Ctrl+Alt+F#** (# 1 to 5).

The following scripting languages have been implemented:

- **NetRexx** (http://www.netrexx.org/): it supports a classic REXX syntax, with no reserved keywords, along with considerable additions to support object-oriented programming in a manner compatible with Java's object model. All existing Java class libraries can be used unchanged and without special setup; at the same time, a Java programmer can opt to just

use the Rexx class from the runtime package for improved string handling in Java syntax source programs.

- **XSLT** (Extensible Stylesheet Language Transformations): is a declarative, XML-based language used for the transformation of XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document may be serialized (output) by the processor in standard XML syntax or in another format, such as HTML or plain text. XSLT is most often used to convert data between different XML schemas or to convert XML data into web pages or PDF documents.

- **Groovy** (http://groovy.codehaus.org): is a dynamic language for the Java Virtual machine. It builds upon the strengths of Java but has additional power features inspired by languages like Python, Ruby and Smalltalk.

- **JavaScript** (sometimes abbreviated JS, not to be confused with Java): is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. Being the language behind popular software such as Firefox it is a familiar and preferred programming tool in the open-source domain.

All the languages have access to the OmegaT object model, with the project as the top object. The following code snippet in groovy for instance scans through all the segments in all files in the current project and, if the translation exists, prints out the source and the target of the segment:

```
files = project.projectFiles;
for (i in 0 ..< files.size())
{
   for (j in 0 ..< files[i].entries.size())
   {
      currSegment = files[i].entries[j];
      if (project.getTranslationInfo(currSegment))
      {
         source = currSegment.getSrcText();
         target = project.getTranslationInfo(currSegment).translation;
         console.println(source + " >>>> " + target);
      }
   }
}
```

Appendix G. OmegaT on the web

1. OmegaT sites and OmegaT SourceForge project

The OmegaT web site [http://www.omegat.org/]contains links to numerous OmegaT resources. User support is provided on a volunteer basis at the OmegaT Yahoo! User Group [http://tech.groups.yahoo.com/group/omegat/]. The FAQ [http://tech.groups.yahoo.com/group/OmegaT/database?method=reportRows&tbl=1] is a good starting point for finding answers to questions you may have. For the latest version of OmegaT, refer to the downloads page at www.omegat.org. You can also file bug reports [https://sourceforge.net/p/omegat/bugs/] and requests for enhancements. [https://sourceforge.net/p/omegat/feature-requests/]

2. Bug reports

Remember that every good bug report needs just three things:

- Steps to reproduce
- What you expected to see
- What you saw instead

More can be found in the Painless Bug Tracking [http://www.joelonsoftware.com/articles/fog0000000029.html] article by Joel Spolsky.

You should add copies of files, portions of the log, screen shots, and anything else that you think will help the developers to find and fix your bug. Note that bug reports and requests for enhancements are publicly visible, so you should not add any sensitive files. If you wish to keep track of what is happening to the report, register as a SourceForge user, login and file a bug report or simply click Monitor at the top of the report.

3. Contributing to OmegaT project

To contribute to OmegaT:

first join the user group [http://tech.groups.yahoo.com/group/OmegaT/] (via web or by sending an email to OmegaT-subscribe@yahoogroups.com [mailto:OmegaT-subscribe@yahoogroups.com]). To get involved in the OmegaT development effort you can join the developer group, via the web or by sending an email to omegat-development-request@lists.sourceforge.net [mailto:omegat-development-request@list.sourceforge.net?subject=subscribe] with "subscribe" as the subject line.

To translate OmegaT's user interface, user manual or other related documents:

first read the Localizing and Customizing OmegaT [http://www.omegat.org/en/howtos/localizing_omegat.php] guide, and join the OmegaT translators' group via web or by sending an email to omegat-l10n-request@lists.sourceforge.net [mailto:omegat-development-request@lists.sourceforge.net?subject=subscribe] with subject "subscribe".

To support the OmegaT project financially

If you would like to help support the continued development of OmegaT, it would be very much appreciated - click on this link to go to the OmegaT PayPal account [https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=9UB6Y2BBF99LL].

Appendix H. Shortcuts customization

1. Shortcuts customization

Most of the items that appear in the main menu can have a new shortcut assigned. You can change the already assigned shortcuts and add new shortcuts by putting a shortcut definition file in your OmegaT preferences folder (see User files location).

The shortcut definition file must be named MainMenuShortcuts.properties and must contain at most one shortcut definition per line. Empty lines are accepted and comment lines should start with "//". Anything after the "//" will be ignored.

Once the MainMenuShortcuts.properties file is modified, OmegaT must be relaunched to take the new shortcuts into account.

The shortcut definition syntax is the following: <menu item code>=<shortcut>, where <*menu item code*> is a code taken from the tables below and <*shortcut*> is a combination of pressed keys specified by the user[1].

<shortcut> must be of the following form: 0 or more <modifier> followed by 0 or 1 <event> followed by 1 <key>, where:

- <modifier> can be: *shift*, *control*, *ctrl*, *meta*[2], *alt*, *altGraph*
- <event> can be: *typed*, *pressed*, *released*
- and <key> can be any key available on your keyboard[3].

For example, in the default OmegaT shortcuts[4], one can find:

- projectOpenMenuItem=ctrl O
- editCreateGlossaryEntryMenuItem=ctrl shift G

The first is the shortcut for Open Project, the second for Create Glossary Entry.

If you want to use **Shift**+**Ctrl**+**O** to open a project, modify your MainMenuShortcuts.properties as follows:

projectOpenMenuItem=shift ctrl O.

If you are on a Mac and you want to add a **Shift**+**Command**+**S** shortcut to Tools → Statistics, add the following line to your MainMenuShortcuts.properties:

toolsShowStatisticsStandardMenuItem=shift meta S

Save then the file and relaunch OmegaT. Your new shortcuts should now appear next to the menu items you have modified. If they do not conflict with system shortcuts, they should be available from within OmegaT.

[1] The full syntax for keystrokes (shortcuts) is defined in the following Java 1.5 documentation from Oracle (bottom of page): Java 1.5 keystrokes shortcuts [http://docs.oracle.com/javase/1.5.0/docs/api/javax/swing/KeyStroke.html]
[2] On the Mac, the modifier *meta* must be used to specify the *command* key.
[3] The possible keyevents (keys) are listed in the following Java 1.5 documentation from Oracle: Java 1.5 keyEvents description [http://docs.oracle.com/javase/1.5.0/docs/api/java/awt/event/KeyEvent.html]
[4] The default OmegaT shortcuts are available from Sourceforge: Default OmegaT Shortcuts [http://omegat.svn.sourceforge.net/viewvc/omegat/branches/release-2-6/src/org/omegat/gui/main/MainMenuShortcuts.properties]

The default OmegaT shortcuts for the Mac are also available from Sourceforge, they all use "meta" instead of "ctrl": Default OmegaT Shortcuts for the Mac [http://omegat.svn.sourceforge.net/viewvc/omegat/branches/release-2-6/src/org/omegat/gui/main/MainMenuShortcuts.mac.properties]

2. Project Menu

Table H.1. Project Menu

Menu Item	Default shortcut	Menu Item Code
New		projectNewMenuItem
Open	**Ctrl+O**	projectOpenMenuItem
Download Team Project		projectTeamNewMenuItem
Import Source Files...		projectImportMenuItem
Import From MediaWiki...		projectWikiImportMenuItem
Reload	**F5**	projectReloadMenuItem
Close	**Ctrl+Shift+W**	projectCloseMenuItem
Save	**Ctrl+S**	projectSaveMenuItem
Create translated documents	**Ctrl+D**	projectCompileMenuItem
Properties...	**Ctrl+E**	projectEditMenuItem
Project Files...	**Ctrl+L**	viewFileListMenuItem
Quit	**Ctrl+Q**	projectExitMenuItem

3. Edit Menu

Table H.2. Edit Menu

Menu Item	Default shortcut	Menu Item Code
Undo Last Action	**Ctrl+Z**	editUndoMenuItem
Redo Last Action	**Ctrl+Y**	editRedoMenuItem
Replace With Match	**Ctrl+R**	editOverwriteTranslationMenuItem
Insert Match	**Ctrl+I**	editInsertTranslationMenuItem
Replace with Machine Translation	**Ctrl+M**	editOverwriteMachineTranslationMen
Replace With Source	**Shift+Ctrl+R**	editOverwriteSourceMenuItem
Insert Source	**Shift+Ctrl+I**	editInsertSourceMenuItem
Insert Source Tags	**Shift+Ctrl+T**	editTagPainterMenuItem
Export Selection	**Shift+Ctrl+C**	editExportSelectionMenuItem
Create Glossary Entry	**Shift+Ctrl+G**	editCreateGlossaryEntryMenuItem
Search Project...	**Ctrl+F**	editFindInProjectMenuItem
Select Fuzzy Match 1	**Ctrl+1**	editSelectFuzzy1MenuItem
Select Fuzzy Match 2	**Ctrl+2**	editSelectFuzzy2MenuItem
Select Fuzzy Match 3	**Ctrl+3**	editSelectFuzzy3MenuItem
Select Fuzzy Match 4	**Ctrl+4**	editSelectFuzzy4MenuItem
Select Fuzzy Match 5	**Ctrl+5**	editSelectFuzzy5MenuItem
Switch Case to/Lower Case		lowerCaseMenuItem
Switch Case to/Upper Case		upperCaseMenuItem
Switch Case to/Title Case		titleCaseMenuItem
Cycle case to...	**Shift+F3**	cycleSwitchCaseMenuItem
Use as Default Translation		editMultipleDefault

Menu Item	Default shortcut	Menu Item Code
Create Alternative Translation		editMultipleAlternate

4. GoTo Menu

Table H.3. GoTo Menu

Menu Item	Default shortcut	Menu Item Code
Next Untranslated Segment	**Ctrl+U**	gotoNextUntranslatedMenuItem
Next Segment	**Ctrl+N** or **Enter** or **Tab**	gotoNextSegmentMenuItem
Previous Segment	**Ctrl+P** or **Ctrl+Enter** or **Ctrl +Tab**	gotoPreviousSegmentMenuItem
Segment number...	**Ctrl+J**	gotoSegmentMenuItem
Next Note		gotoNextNoteMenuItem
Previous Note		gotoPreviousNoteMenuItem
Forward in history...	**Ctrl+Shift+N**	gotoHistoryForwardMenuItem
Back in history...	**Ctrl+Shift+P**	gotoHistoryBackMenuItem

5. View Menu

Table H.4. View Menu

Menu Item	Default shortcut	Menu Item Code
Mark Translated Segments		viewMarkTranslatedSegmentsCheckB
Mark Untranslated Segments		viewMarkUntranslatedSegmentsChec
Display Source Segments		viewDisplaySegmentSourceCheckBox
Mark Non-Unique Segments		viewMarkNonUniqueSegmentsCheckB
Mark Segments with Notes		viewMarkNotedSegmentsCheckBoxM
Mark Non-breakable Spaces		viewMarkNBSPCheckBoxMenuItem
Mark Whitespace		viewMarkWhitespaceCheckBoxMenuI
Mark Bidirectional Algorithm Control Characters		viewMarkBidiCheckBoxMenuItem
Modification Info/Display None		viewDisplayModificationInfoNoneRad
Modification Info/Display Selected		viewDisplayModificationInfoSelectedR
Modification Info/Display All		viewDisplayModificationInfoAllRadioB

6. Tools Menu

Table H.5. Tools Menu

Menu Item	Default shortcut	Menu Item Code
Validate Tags	**Ctrl +T**	toolsValidateTagsMenuItem
Statistics		toolsShowStatisticsStandardMenuIter
Match Statistics		toolsShowStatisticsMatchesMenuItem

7. Options Menu

Table H.6. Options Menu

Menu Item	Default shortcut	Menu Item Code
Use TAB To Advance		optionsTabAdvanceCheckBoxMenuIte
Always Confirm Quit		optionsAlwaysConfirmQuitCheckBoxM
Machine Translate		
TransTips/Enable Transtips		optionsTransTipsEnableMenuItem
TransTips/Exact Match		optionsTransTipsExactMatchMenuIter
Font...		optionsFontSelectionMenuItem
File Filters...		optionsSetupFileFiltersMenuItem
Segmentation...		optionsSentsegMenuItem
Spell checking...		optionsSpellCheckMenuItem
Editing Behavior...		optionsWorkflowMenuItem
Tag Validation...		optionsTagValidationMenuItem
Team...		optionsTeamMenuItem
External TMXs...		optionsExtTMXMenuItem
View...		optionsViewOptionsMenuItem
Saving and Output...		optionsSaveOptionsMenuItem
Proxy login...		optionsViewOptionsMenuLoginItem
Restore Main Window		optionsRestoreGUIMenuItem

8. Help Menu

Table H.7. Help Menu

Menu Item	Default shortcut	Menu Item Code
User Manual...	**F1**	helpContentsMenuItem
About...		helpAboutMenuItem

Appendix I. Legal notices

1. For the documentation

Copyright

The documentation distributed with OmegaT includes the User Manual and the readme.txt document. The documentation is Copyright ©2013 Vito Smolej. The author of the Chapter *Learn to use OmegaT in 5 minutes!* is Samuel Murray, Copyright ©2005-2012.

Distribution and modifications

The documentation is a free document; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (if you prefer) any later version.

Warranty

The documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

2. For the application

Copyright

OmegaT is Copyright © 2000-2013 Keith Godfrey, Zoltan Bartko, Volker Berlin, Didier Briel, Kim Bruning, Alex Buloichik, Thomas Cordonnier, Sandra Jean Chua, Martin Fleurke, Wildrich Fourie, Phillip Hall, Jean-Christophe Helary, Thomas Huriaux, Hans-Peter Jacobs, Ibai Lakunza Velasco, Guido Leenders, Aaron Madlon-Kay, Fabián Mandelbaum, John Moran, Maxym Mykhalchuk, Arno Peters, Henry Pijffers, Briac Pilpré, Tiago Saboga, Andrzej Sawuła, Benjamin Siband, Yu Tang, Rashid Umarov, Antonio Vilei, Martin Wunderlich and Michael Zakharov.

Distribution and modifications

OmegaT is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (if you prefer) any later version.

Warranty

OmegaT is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more detail.

Appendix J. Acknowledgements

1. Thank you all!

Whatever the inconsistencies, omissions and straightforward errors you may find in the present version, I declare them all my own. This manual, however, would not be possible without the help and support from a number of people. Explicit thanks to:

- Marc Prior: correcting my first draft was an act of love for OmegaT and the English language.
- Didier Briel: I could not do without Didier's patient and persistent help with DocBook intricacies. Not to mention his care and diligence, keeping repositories intact and in good order.
- Samuel Murray: for the introductory chapter "Learn to use OmegaT in 5 minutes".
- Will Helton: his final reading of the draft has spared me a lot of embarrassment. One could only wonder, how many the and a prepositions would still be missing without his invaluable help.
- Jean-Christophe Helary: special thanks to JC for his concise description of OmegaT run, command line parameters and all other details, I have yet to notice.
- Last but not least: my thanks to all the contributors to OmegaT documentation tracker [https://sourceforge.net/p/omegat/documentation/] for all the inconsistencies found in the previous versions of the documentation. Keep up your good work!

Index

C

Comments
  Comments pane, 21
Customizing OmegaT
  Linux, 6
  OS X
    Launch parameters, 7

D

Dictionaries, 74
  Britannica, 74
  Downloading and installing, 74
  Longman, 74
  Merriam Webster, 74
    (see also Dictionaries)
  Problems with, 75
  StarDict, 74
  Webster, 74

E

Encoding
  Central and Eastern European, 54
  Plain text files, 54
  Unicode, 54
  Western, 54

F

File filters
  Dialog, 38, 41
  Editing, 40
  File type and name pattern, 40
  global vs project file filters, 36
  Options, 38
  Project specific file filters, 38
  Source, target - encoding, 40
File formats
  formatted, 48
    (see also Source files)
  Unformatted, 48
    (see also Source files)

G

Glossaries, 20, 76
  Creating a glossary, 77
  File format, 77
  Glossary pane
    multiple-words entries, 77
  Location of the writable glossary file, 37, 78
  Microsoft Terminology collection, 77
  Problems with glossaries, 78
  TBX format, 77
  Trados MultiTerm, 78
Glossaries, Glossary pane, 76

I

Installing OmegaT
  Linux, 6
  OS X, 7
  Other systems, 8, 8
  Windows, 5
ISO language codes, 89

K

Keyboard shortcuts, 31
  Editing, 32
  Goto, 32, 33
  Other, 33
  Project, 31

L

Languages, 89
Legal notices, 112
  For the application, 112
  For the documentation, 112
Lucene (see Tokenizer)

M

Machine Translation, 80
  Apertium, 81
  Belazar, 81
  Google Translate, 80
  Introduction, 80
  Troubleshooting, 82
Match Statistics, 23
  (see also Menu Tools)
Matches
  Matches pane - figure, 18
  Matches pane setup - figure, 19
  Matches statistics, 29
Menu, 25
  Edit, 26
  Goto, 28
  Help, 31
  Options, 30
    Editing behavior..., 52
  Project, 25
  Tools, 29
  View, 29
Menu Help
  Help browser, 24
  User Manual..., 24
Menu Options
  Editing Behavior, 16
  Editing behaviour
    Converting numbers, 53
    Empty translation, 52
    Exporting the current segment, 53
    Inserting fuzzy matches, 52
    Segments with alternative translation, 53
    Translation equal to source, 53
  File filters, 15
  Font..., 87
  Segmentation, 16
  Spell checking, 83
Menu Project
  New..., 3

Properties, 15
Menu Tools
Match statistics, 15
statistics, 15
Miscellanea, 86
Automatic aligner for Java properties, 87
Font settings, 87
OmegaT console mode, 86
Preventing data Loss, 88

O

OmegaT
Team projects, 96
(see also Team projects)
OmegaT console mode, 86
(see also Miscellanea)
OmegaT on the web, 107
Contributing to OmegaT, 107
Development, Localizing, 107
Donating to OmegaT, 107
Financial support, 107
Reporting bugs, 107
SourceForge Project, 107
OmegaT windows, 17
(see also Windows and panes in OmegaT)
Restoring to factory setup, 17

P

Plugins
LanguageTool, 104
Scripting, 105
Tokenizer, 102
Project
Autopropagation option, 36
Changing the folder structure, 37
Create / open new, 3
Enable Sentence-level segmentation, 36
Match statistics, 29
Options, 65
Pretranslation, 60
Project management shortcuts, 31
Properties, 34, 65, 89
(see also Languages)
Remove tags option, 36
Statistics, 29
Project files
Application files, 46
File omegat.project, 44
Glossary subfolder, 76
ignored_words and learned_words, 44
Source subfolder, 45
statistics file, 44
Subfolder omegat, 44
Target subfolder, 45
Translation project files, 43
User files, 77
(see also Glossaries)
User settings files, 45
Project Settings
Editing Behavior, 16
File filters, 15
Font, 15
Segmentation, 16

R

Regular expressions, 70
(see also Searching)
(see also Segmentation)
Examples of use, 72
Tools, 72
Right to left languages, 50
Creating RTL target files, 50
Creating RTL target text, 50
Mixing RTL and LTR strings, 50
OmegaT tags in RTL languages, 50
Target files, 50
Running OmegaT
Building OmegaT from source, 14
Command line launching, 9
Command line mode, 11
Launch script arguments, 10
Font antialiasing, 10
Google Translate V2, 11
Memory assignment, 11
Microsoft Translator, 11
no-team option, 11
Proxy host IP address, 11
Proxy host port number, 11
User country, 10
User interface language, 10
Linux, 7
OS X, 7
Other systems, 8
Using Java Web Start, 9
Windows, 5
INI file, 5

S

Search window
Example, 67
Searches, 67
Advanced Options, 68
Methods and options, 68
Using wild cards, 68
Segment marker, 18
Segmentation
Creating a new rule, 66
(see also Regular expressions)
Examples, 66
global vs project rules, 36
Rules, 65
Break rule, 65
Exception rule, 65
Rules priority, 66
Sentence level segmentation, 65
Source level segmentation, 65
Shortcuts
Case selection - Shift+F3, 27
Copy text - Ctrl+C, 23
copy text - Ctrl+C, 24

Customization, 108
Help - F1, 15, 24
Insert text - Ctrl+I, 18
Machine Translate - Ctrl+M, 81
Paste text - Ctrl+V, 23, 24
Project files list - Ctrl+L, 21
Project properties - Ctrl+E, 15, 75
Replace text - Ctrl+R, 18
Search - Ctrl+F, 67
Select All - Ctrl+A, 23, 24
Selecting the match - Ctrl+N, 18
Tag validation - Ctrl+T, 15, 57
Shortcuts Customization
View Menu, 110
Source files
Adding files to the project, 22
Encoding, 40, 54
File formats, 48
File type and name pattern, 40
Formatted text, 55
Formatted text files, 48
Mixing RTL and LTR strings, 50
Other file formats, 49
Plain text files, 48, 54
PO as bilingual files, 48
Right to left languages, 50
Translating updated source, 19
Spell checker, 83
Hints, 84
Spell checker setup, 83
Statistics, 23
(see also Menu Tools)
Stemmer (see Tokenizer)

T

Tag validation, 22
(see also Shortcuts, Tags)
Window - figure, 22
(see also Tags)
Tags, 55
Duplication, 55
Group deletion, 56
Group nesting, 56
Group overlapping, 56
Group validation, 57
Hints, 58
Inserting source tags, 27
Naming, 55
Numbering, 55
Operations, 55
Pairs and singles, 55
Target files
Encoding, 40
File conversion tools, 49
File formats, 48
Filenames, 41
Formatted text, 55
Formatted text files, 48
(see also Tagged text)
Mixing RTL and LTR strings, 50
Other file formats, 49
Plain text files, 48
Right to left languages, 50
Team projects
Creating SVN repository, 97
Subversion, 96
TMX (see Translation memories)
Translation memories, 59
Alternative language pairs, 63
Backup, 60
compressed, 60
Importing and exporting, 62
Language, 61
Matches, 19
multilingual, handling of, 61
Orphan segments, 19, 61
PO and OKAPI TTX files, 63
(see also Translation memories Subfolder tm/auto)
Project main folder, 59
Pseudotranslation, 64
Reusing translation memories, 61
Sharing, 63
(see also Project,Download Team Project...)
Subfolder omegat, 59
(see also Project files)
Subfolder tm, 59
(see also Project files)
Subfolder tm/auto, 60
(see also Project files)
Subfolders tm/penalty-xxx, 60
(see also Project files)
Upgrading to sentence segmentation, 64

U

Upgrading OmegaT
Windows, 5
User Interface
Main OmegaT window, 15
Match pane setup, 19
User interface
Other windows, 15
Settings dialogs, 15
(see also Project Settings)

W

Windows and panes in OmegaT
Counters, 17
Dictionary pane, 20
Editor pane, 18
Fuzzy matches pane, 18
Customizing, 19
Glossary pane, 20, 76
Machine Translation pane, 21
Main window, 16
Matches pane - figure, 18
Matches pane setup - figure, 19
Multiple Translations pane, 20
Pane widgets, 16
Project files, 21

Search pane, 22
Tag validation, 22
(see also Tags)